

BIOGRAPHICAL INFORMATION

Kurt Buehler
Vice President and Chief Technology Officer
Open GIS Consortium, Inc.

Specific Responsibilities

Mr Buehler, a co-founder of the Open GIS Consortium (OGC), presently acts as Vice President and Chief Technology Officer of OGC. He has been in this role since 1995. He is responsible for all technical aspects of OGC activities. As one of two permanent Directors and in his role as Vice President he is also involved in all operational aspects of OGC.

Past Experience

Prior to joining OGC, Mr. Buehler was employed as Team Leader of the Spatial Analysis and Systems Team as the U.S. Army Construction Engineering Research Laboratories (USACERL). In his role as Team Leader he managed the development and release processes of the GRASS Geographic Information System and developed and maintained parts of the GRASS GIS.

Mr. Buehler's time at USACERL was preceded by two year long research appointments, the first as a Professional Research Assistant at the Center for Advanced Decision Support for Water and Environmental Systems at the University of Colorado at Boulder and the second as a Research Engineer with the Indiana Water Resources Research Center at Purdue University.

Educational Information

B.S.C.E. – Civil Engineering, Purdue University

M.S.C.E. – Civil Engineering, Purdue University

Professional Memberships

GITA

GEOSPATIAL INTEROPERABILITY BY DESIGN

Kurt Buehler¹

Open GIS Consortium, Inc.

4899 N Old State Road 37, Bloomington, Indiana 47408

ABSTRACT

Interoperability doesn't happen by chance. Interoperability happens through, design and planning. Systems architecture is a key part of any interoperability solution. This paper outlines the OpenGIS[®] Reference Model (ORM). The ORM is an interoperability framework that can be used to construct other, more specific, domain or application interoperability frameworks and architectures. A few examples will be provided of the latter type of framework. Finally, the author will detail the steps required to reach an architecture that is specific enough for a procurement that includes interoperability by design.

INTRODUCTION

Systems (or information technology (IT)) architecture is a key part of any interoperability solution. The critical importance of IT architecture became increasingly appreciated in the 1990s, and a number a number of significant books, standards, and guidelines concerning good practice have been written and established. One of the most important developments was an international standard called the Reference Model of Open Distributed Processing (RM-ODP). The OpenGIS Reference Model (ORM) is an architectural framework that follows RM-ODP and is intended to support the development of domain or application or enterprise specific architectures (also using RM-ODP). This paper introduces both RM-ODP and ORM and then describes how to use them both by providing recipe-based examples. There is no implication by this simplistic approach minimizes the effort required to architect a major system using these principles. Rather, by using an appropriate set of tools, the architecture design effort will progress more easily, The use of a reference framework such as the ORM allows systems engineers and developers to more effectively design and deploy applications based on the concepts of interoperability.

THE REFERENCE MODEL OF OPEN DISTRIBUTED PROCESSING

RM-ODP is an international standard (ISO/IEC 10746). RM-ODP is a framework for defining architectures. RM-ODP's organizing framework is built around five relatively orthogonal but consistently defined viewpoints:

- An *Enterprise Viewpoint*, which defines the business perspective of the system: its purpose and scope; its operations, roles, and policies; including such terms as enterprise object, role, interaction, process, take, and policy.
- An *Information Viewpoint*, which defines the semantics of and relations among the system's information elements; including such terms as information object, static schema, dynamic schema, and invariant schema.
- A *Computational Viewpoint*, which defines the system functionality and its partitioning into components; including such terms as computational object, interface, interaction, client, server, producer, and consumer.
- An *Engineering Viewpoint*, which defines the system's distribution; including such terms as engineering object, channel, node, protocol, interface, replicate, and checkpoint/recovery.
- A *Technology Viewpoint*, which defines the system's technology and products, such as commercial packages and developed code; including conformance points for testing the conformance of code to its specification.

¹ Portions of this paper were adapted from an article to appear in the March 2004 issues of Geointel magazine and are printed here by permission of AdvanStar communications.

These viewpoints provide the architect with “bins” for defining and describing all of the major architectural issues in an organized and non-overlapping way. This fact is extremely useful and results in good architecture because issues are not lost in a morass of interwoven threads. Within the RM-ODP framework, consistency is provided by sets of basic rules, object model rules, structuring rules, specification rules, and conformance rules. The “openness” of RM-ODP is facilitated by a set of *distribution transparencies* covering access, failure, location, migration, relocation, replication, persistence, and transaction consistency. These basically hide implementation details, enabling multiple product implementations to be used interchangeably. The author highly recommends a book by Janis R. Putman, entitled “Architecting with RM-ODP,” that provides an excellent basis for the practical use of RM-ODP.

Before looking at how we use the RM-ODP as a tool, we look at one other tool, Model Driven Architecture (MDA).

Model Driven Architecture for Platform Independence

Model Driven Architecture (MDA) is an approach for designing and developing complex systems leveraging the capabilities of UML (Unified Modeling Language) models to manage complexity. In the pure form of MDA, the UML models themselves are sufficient to build a system. A system is initially designed using UML models to capture operational, computational, and information design decisions. Software tools then convert the UML models into executable code suitable for the target processing environment.

In reality, executable UML is rarely possible. For most systems, the value of MDA is that it enables the architect to formally define the conceptual design for a system while leaving developers the freedom to implement that design using the technology appropriate for their specific requirements. Since the implementations are built from a common UML base, mediation services² can integrate the implementations into the greater enterprise.

THE OPENGIS REFERENCE MODEL

The OpenGIS Reference Model (ORM) provides an architecture framework for the ongoing work of the Open GIS Consortium (OGC). Further, the ORM provides a framework for the OGC Technical Baseline. The OGC Technical Baseline consists of the currently approved OpenGIS Specifications as well as for a number of candidate specifications that are currently in progress.

The ORM has the following purposes (including, in italics, the most relevant to the discussion in this paper):

- Provides a foundation for coordination and understanding (both internal and external to OGC) of ongoing OGC activities and the Technical Baseline;
- Update/Replacement of parts of the 1998 OpenGIS Guide;
- *Describes the OGC requirements baseline for geospatial interoperability;*
- *Describes the OGC architecture framework through a series of non-overlapping viewpoints: including existing and future elements;*
- *Regularize the development of domain-specific interoperability architectures by providing examples.*

The ORM provides architects of systems with significant geospatial elements with a tool to kick-start their architecting using RM-ODP. The main reason that the ORM is useful in this context is that it uses RM-ODP as a model for its expression and so it has all of the five main Viewpoints within its content. Architects can therefore reuse many of the elements within it and will be in line with the mainstream of the geospatial standards community. This section will provide a bit of an overview of the ORM in terms of its five viewpoints.

² A mediation service provides mechanisms for “homogenizing,” or resolving the data mismatches and process sequence mismatches of heterogeneous Web services.

Enterprise Viewpoint

The Enterprise Viewpoint in the ORM briefly describes the OGC enterprise in its business perspective, purpose, scope and policies. After highlighting the role of geospatial location as a fundamental information ingredient, the Enterprise Viewpoint provides a representative value chain of geospatial information within an enterprise or an information community. The viewpoint concludes by highlighting the major requirements on OGC technology as derived from the described value chain. Because OGC is a specification development organization, our Enterprise Viewpoint will not likely be very useful to architects, except to indicate, by example, the kind of documentation that fits into the viewpoint.

Information Viewpoint

The Information Viewpoint in the ORM is concerned with the semantics of information and information processing. The Information Viewpoint defines conceptual schemas for geospatial information and methods for defining application schemas. The conceptual, or base, schemas are formal descriptions of the model of any geospatial information. Application schemas are information models for a specific information community. Applications schemas are built from the conceptual schemas.

The Information Viewpoint begins with a discussion of geographic features as the basic concept of modeling geospatial information, followed by several sections presenting the conceptual schemas, and closes with a single section on application schemas.

This Information Viewpoint is an extremely useful element to organizations wishing to use the ORM along with RM-ODP because it defines an extremely wide range of geospatially-related information elements. This fact means that relevant portion of the ORM Information Viewpoint could be adapted directly to the Information Viewpoint document for the system being designed.

Computational Viewpoint

The Computational Viewpoint is concerned with the functional decomposition of the system into a set of services that interact at interfaces. This viewpoint captures the details of these components and interfaces without regard to distribution. Accordingly, this section of the ORM first defines the core concepts of services, interfaces and operations (and the relationships amongst these concepts) and then describes the Publish/Find/Bind pattern that represents the interactions among OGC services. The Computational Viewpoint of the ORM also includes examples of service classification and a description of the OGC Service Framework. Example service applications where services are combined or chained in support of decision-making are included at the end of this section.

Once again, as with the Information Viewpoint, the Computational Viewpoint is an extremely useful element to organizations planning to use the ORM along with RM-ODP because it defines an extremely wide range of geospatially-related service elements. This fact means that relevant portion of the ORM Computational Viewpoint could be adapted directly to the Computational Viewpoint document for the system being designed.

Engineering Viewpoint

The Enterprise, Information, and Computation viewpoints describe a system in terms of its purposes, its content, and its functions. The Engineering viewpoint relates these to specific components linked by a communications network. This viewpoint is concerned primarily with the interaction between distinct computational objects³: its chief concerns are communication, computing systems, software processes and the clustering of computational functions at physical nodes of a communications network. The engineering viewpoint also provides terms for assessing the “transparency” of a system of networked components – that is, how well each piece works without detailed knowledge of the computational infrastructure. The engineering viewpoint can be described in terms of UML collaboration diagrams and deployment diagrams.⁴

³ <http://www.cs.tcd.ie/synapses/public/deliverables/part1.pdf>

⁴ http://www.informatics.sintef.no/~jol/ODP/rmodp_overview.html

In its role as a reference architecture and framework, the ORM is not specific to any particular system implementation or DCP. therefore, the Engineering Viewpoint of the ORM covers very general issues related to more specific Engineering Viewpoints. Topics covered include, multi-tier architectures and their constituent elements (e.g., clients, application servers, data servers), bridging multiple networks (e.g., the Internet and a wireless operator network), and the topic of distribution transparencies.

This section of the ORM should be used to outline the types of issues to be covered in an Engineering Viewpoint, but will likely not be as directly useful as some of the other viewpoints.

Technology Viewpoint

The Technology Viewpoint is concerned with the underlying infrastructure in a distributed system. It describes the hardware and software components. The infrastructure, which may be provided by a Distributed Computing Platform (DCP), allows objects to interoperate across computer networks, hardware platforms, operating systems and programming languages.

This section in the ORM provides an overview of OGC's multiple DCP implementation approach, followed by a list of the current encodings used within OGC, including XML, imagery and well-known binary and text encodings. The Technology Viewpoint also summarizes the technologies chosen for the Web Services and the OpenGIS Location Services (OLS) platform implementations.

This section of the ORM should prove to be an extremely useful element to organizations wishing to use the ORM along with RM-ODP because it defines the detailed technology specification that architects could put into procurements for system elements. This fact means that relevant portion of the ORM Technology Viewpoint could be adapted directly to the Technology Viewpoint document for the system being designed.

The Examples

The ORM describes three example scenarios to provide the reader with a better understanding of how to apply the ORM. These include an Information Producer, and Information Broker, and an Information Consumer scenario. We briefly overview these scenarios here and direct the reader to the ORM for more detail.

Information Producer Scenario

The first scenario focuses on the production of a standard product called Foundation Feature Information (FFI), from an assortment of sources, at a Civilian Mapping Agency.

This formal scenario name is the "Base Map Production Unit (BMPU) of the Average Civilian Mapping Agency (ACMA)". The scenario is called the *Production Scenario* for short. The motto of the ACMA is "FFI Production R Us."

From the Enterprise Viewpoint, this scenario encompasses the central reason for the existence of the Civilian Mapping Agency: the production of timely, accurate, and useful Feature Foundation Information to include elevation information, imagery information, and feature information. A variety of input, resources, and control parameters make the production goals possible.

The scenario focus is on three major steps: (1) the establishment of image control so that stereoscopic and photogrammetric technology can support the discovery of features and the extraction of feature geometry, attributes, and relationships in support of FFI production, (2) the extraction and/or update of new features, and (3) the fusion of new features into legacy FFI to create new or updated information conforming to FFI specifications.

Information Broker Scenario

The Greater Chitroit Council of Governments (GCCOG) is cooperating with the Greater Chitroit Chamber of Commerce and Board of Realtors to solicit plans from major developers for a large multi-purpose development according to the GCCOG Long Range Development Plan. The intentions are to increase development of light industrial, retail, and single family housing in areas of the region that are in need. Several alternatives are proposed.

The use case starts when the GCCOG receives a Regional development plan from a major developer with alternative locations for implementing the plan. GCCOG evaluates the plan against a set of econometric, traffic, and environmental models. To evaluate the proposals, the GCCOG has adopted an econometric model and a traffic prediction model. Each member of the GCCOG will use visualization and portrayal tools and searches to compare the outcomes of various alternatives. Similarly, the GCCOG will be using environmental parameter data and environmental models to determine impacts of the development. Additionally, the plan is evaluated against GCCOG long-range development plan.

The GCCOG has retained GSTIM, Inc, to develop a State Spatial Data Infrastructure that is interoperable with the National Spatial Data Infrastructure. This RSDI will link local and state information providers together (like the participants in the NSDI) in an environment where data and services can be shared.

Information Consumer Scenario

A consumer is traveling to a dental appointment and the wireless service provider is offering travel assistance through a Personal Navigation Service. At the onset, the subscriber only knows the phone number of the dentist. The consumer is equipped with a location-enabled cell phone, sees that a major storm is approaching, and wants to obtain routing information from the provider. The routing information is updated to reflect current traffic and weather conditions. The scenario relates to the consumers use of the Personal Navigation Service to obtain weather and traffic sensitive directions to his dentist.

APPLICATION OF THE TOOLS

We will provide a basic recipe for the application of the RM-ODP and the ORM in developing architecture with interoperability built-in and that will support procurement. A successful procurement is facilitated by unambiguous expression of high level requirements (Enterprise Viewpoint), the information elements (Information Viewpoint), the functionality expressed as components (Computational Viewpoint), the distribution architecture in terms of clients, servers, and networks (Engineering Viewpoint), and finally, the existing and desire technologies, hardware, software packages, operating systems and how these relate to standards and specifications for each (Technology Viewpoint).

The Enterprise Viewpoint is where business processes are captured. We start by modeling a set of business processes. These processes are then decomposed into their atomic actions. These primitive use-cases are used to capture these atomic actions.

The next step is to identify the information needs of the system. Like the atomic actions, information needs are identified in the business processes and decomposed into primitive information types. Once completed, we have a language consisting of nouns (primitive information types) and verbs (primitive use-cases) that are sufficient to express all known business processes. The next question is – what can we do with this “language”?

The Computational Viewpoint is used to expand the verbs of the language. Each primitive use-case identifies an information processing capability that the system must support. Each one of these capabilities corresponds to an information processing service. The behavior of that service is defined by its associated use case. Additional detail is provided to identify the interfaces, operations, and parameters needed to implement that behavior. This provides us with sufficient information to create UML models of the services. Elements from the ORM Computational Viewpoint and specific UML models from OGC documentation related to this viewpoint can be brought in at this point.

The Information Viewpoint is used to expand the meaning of the nouns in the language. Each primitive use-case identifies the information exchanged as part of each transaction. UML models of the information types are developed to capture the specific data elements necessary to convey the required information. Where possible, existing information standards should be leveraged and elements from the ORM Information Viewpoint can be brought in at this point.

This set of Enterprise, Computational, and Information views comprise the *abstract framework*. This framework establishes a coherent picture of the range of business processes that can be performed as well as the computational and information resources needed to support those processes. Additional business processes, computational, and information resources can be added to these models as needed.

The next step is to understand the technical constraints that apply to the system. These constraints include available hardware and communications resources, operating systems, and perhaps most important, the Distributed Computing Platform (DCP). In order for two components to interoperate, there must be a set of standard protocols, encodings, and supporting services to enable that interaction. This collection of protocols, encodings, and services constitute a Distributed Computing Platform (DCP). Different DCPs have different strengths and weaknesses. The Java 2 Enterprise Edition (J2EE), for example, is a robust distributed computing platform that is suitable for high reliability and high performance applications. At the other end of the spectrum, REST is a minimalist approach for exchanging request/response messages that is ideal for situations where middleware vendor and version incompatibilities can be expected. System designers examine the mission objectives of the system and select the DCP(s) that are appropriate. This collection of constraints is documented in the Engineering Viewpoint.

The final step is to generate implementations of the services and information models defined in the Computational and Information views. Starting with the UML models, scripts and tools specific for the target environment (as defined in the Engineering Viewpoint) are employed to generate interface code (if integrating or developing internally) or to specify for a procurement (if contracting for integration and/or development). While MDA techniques may not provide a final implementation, the code generated goes a long way to reaching that final set and provides assurance that the interfaces will be consistent with the abstract views.

CONCLUSIONS

The RM-ODP provides important guidance to information system architects who are designing service architectures for enterprise information systems. MDA provides an additional layer of abstraction that enables an architect to define in a formal manner the conceptual design for a system while leaving developers the freedom to implement that design using the technology appropriate for their specific requirements.

The existence of a comprehensive open services framework (the ORM) in the geoprocessing industry enables system architects to detail spatial information requirements in the Enterprise Viewpoint with confidence that the requirements can be met by standards-based commercial off-the-shelf (SCOTS) component products. The actual components are specified in the Computational Viewpoint. The architect specifies in the Information View the data models (for which standards may also be available) that are also necessary to meet these information requirements.