A spatial data infrastructure model from the computational viewpoint

Antony K Cooper[a*], Harold Moellering[b], Jan Hjelmager[c], Petr Rapant[d], Tatiana Delgado[e], Dominique Laurent[f], Serena Coetzee[g], David M Danko[h], Ulrich Düren[i], Adam Iwaniak[j], Jean Brodeur[k], Paloma Abad[l], Michel Huet[m] and Abbas Rajabifard[n]

[a]Built Environment, CSIR, PO Box 395, Pretoria, 0001, South Africa

[b]Department of Geography, Ohio State University, Columbus, OH, 43210, USA

[c]Kort & Matrikelstyrelsen, Rentemestervej 8, Copenhagen, DK-2400 NV, Denmark

[d]IT4Innovations, VSB-Technical University of Ostrava, 17. listopadu 15, Ostrava-Poruba, Czech Republic, and Institute of Geoinfomatics, Faculty of Mining and Geology, VSB-Technical University of Ostrava, 17. Listopadu 15, Ostrava-Poruba, Czech Republic

[e]Department of Information Systems, Industrial Engineering Faculty, ISPJAE University, 119#11901 Marianao. La Habana, Cuba

[f]Institut Géographique National, 4 Rue Pasteur, 94165 Saint Mandé, France

[g]Centre for Geoinformation Science, University of Pretoria, Pretoria, 0002, South Africa

[h]Environmental Systems Research Institute, Inc, 8620 Westwood Center Drive, Vienna, VA 22182-2214, USA

[i]Landesvermessungsamt Nordrhein-Westfalen, Muffendorfer Strasse 19-21, 53177 Bonn, Germany

[j]Institute of Geodesy and Geoinformatics, Wrocław University of Environmental and Life Sciences, ul. Grunwaldzka 53, 50-357 Wrocław Poland

[k]Natural Resources Canada, Centre for Topographic Information, 2144-010 King West Street, Sherbrooke, Quebec, Canada J1J 2E8

[l]Instituto Geográfico Nacional, General Ibañez de Ibero 3, 28003 Madrid, Spain

[m]International Hydrographic Bureau, 4 quai Antoine 1er, Monaco

[n]Department of Geomatics, University of Melbourne, Melbourne, Victoria 3010, Australia

[*] Corresponding author. Email: acooper@csir.co.za

The Commission on Geoinformation Infrastructures and Standards of the International Cartographic Association (ICA) is working on defining models of spatial data infrastructures (SDI). SDI models from the enterprise and information viewpoints of the Reference Model for Open Distributed Processing (RM-ODP) have already been presented. Our model from the computational viewpoint identifies the main computational objects of an SDI and their interfaces, which are modelled using Unified Modelling Language (UML) component diagrams. Presented here is the first comprehensive SDI model from the computational viewpoint, which enhances the understanding of the computational objects and their interactions in an SDI. This viewpoint complements the previous two and together, the three viewpoints contribute towards a more holistic interpretation of an SDI, which is independent of specific SDI legislation, technology and implementations. For the computational viewpoint, we identified six computational objects, *SDI Registry*, *SDI Data*, *SDI Processing*, *SDI Application*, *SDI Portrayal* and *SDI Management,* and their provided and required interfaces. We describe the interactions of the computational objects in stakeholder activities and the roles they play in the different processes of SDI development and use, which we identified as *Initiation*, *Creation*, *Management*, *Manipulation*, *Access*, *Processing*, *Evaluation* and *Liaison*. Two tables summarise the SDI services that are provided by computational objects for stakeholder activities and SDI processes.

Keywords: spatial data infrastructure; SDI; analytical cartography; reference model; computational viewpoint; unified modelling language; UML

1. Introduction

Over many decades, spatial data scientists in many parts of the world have worked to develop mechanisms to share various kinds of scientific spatial data. More recently, efforts have been organized to design and build spatial data infrastructures (SDI) at the global, regional, national and local levels. There are two major facets to this effort: organizational and technical. The organizational side is concerned with various groups developing agreements, protocols and policy strategies for the SDI, see: Groot & McLaughlin (2000), Masser (2005), Rajabifard *et al* (2006), and Delgado (2005). Another aspect of the organizational side is the building and maintenance of the organizational structures to maintain the elements of the SDI, as well as running the daily SDI business.

The second facet of the SDI effort is of a technical nature: designing and implementing the mechanisms and networks that will bring the SDI to reality. This article focuses on the technical facet. More specifically, it discusses the contribution of the Commission on Geoinformation Infrastructure and Standards of the International Cartographic Association (ICA) (hereafter called *the Commission*) by modelling the SDI through the prism of the various model views of the Reference Model for Open Distributed Processing (RM-ODP) (ISO/IEC 10746-1:1998), using the Unified Modelling Language (UML) (ISO/IEC 19501:2005, Object Management Group 2011). The work discussed here operates in a broader scientific milieu of geographical information science and analytical cartography, e.g. see Moellering (2000).

Previous work by the Commission described SDI models from the enterprise and information viewpoints (Hjelmager *et al* 2005, Hjelmager *et al* 2008). In this article, the Commission's SDI model from the computational viewpoint is presented. The computational viewpoint is a functional decomposition of a system into a set of objects that interact at interfaces – enabling system distribution (ISO/IEC 10746-1:1998). This article consolidates and refines earlier work. The initial SDI model from a computational viewpoint identified computational objects and briefly described them (Cooper *et al* 2007). Earlier work identified the processes of SDI development and use (Cooper *et al* 2009). In this article the descriptions are refined and examples are added to improve the understanding. We also consolidate earlier work by describing how stakeholders (identified in the enterprise viewpoint) interact through interfaces (identified in the computational viewpoint) during the activities associated with the information classes (identified in the information viewpoint).

The remainder of the article is structured as follows: section 2 provides a short overview of previous work by the Commission leading up to this article, as well as other work that is related to ours. Section 3 describes an SDI from the computational viewpoint. Section 4 links the enterprise, information and computational viewpoints by showing how stakeholders identified in the enterprise viewpoint interact through the interfaces identified in the computational viewpoint during the activities identified in the information viewpoint. Section 5 identifies the different processes of SDI development and use, and then shows how SDI services are provided by computational objects for these SDI processes. Examples from the European SDI, INSPIRE (INfrastructure for SPatial InfoRmation in Europe), are included to illustrate the service needs for processes. Two tables summarize the specific SDI services for the stakeholders provided by each computational object for each activity (Table 1 in section 4) and process (Table 2 in section 5) respectively. Section 7 evaluates the SDI model from a computational viewpoint and provides conclusions on this work.

2. Previous and related work

2.1 SDI modelling framework

Business and enterprise systems are becoming more and more flexible and increasingly dependent on local and global communication networks. As a result, the software systems have to become more modular and distributed throughout these networks. The design and installation of such distributed systems is a complex task, which requires comprehensive conceptual work before beginning the implementation. The International Standard ISO/IEC 10746-1:1998, *Information technology – Open Distributed Processing – Reference model: Overview*, aids in this challenge by providing a framework for the design and description of distributed software and information systems.

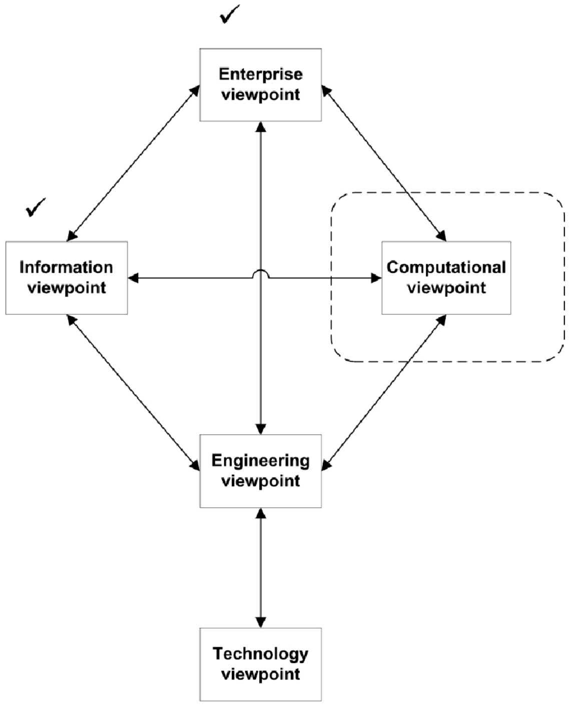

Figure 1: The five RM-ODP Viewpoints and their dependencies (from Hjelmager *et al* 2008)

RM-ODP defines five *viewpoints* for the design and description of distributed software and information systems: the *enterprise viewpoint* (purpose, scope and policies for the system); the *information viewpoint* (semantics of information and information processing incorporated into the system); the *computational viewpoint* (functional decomposition of the system into a set of objects that interact at interfaces); the *engineering viewpoint* (mechanisms and functions required to support distributed interaction between objects within the system); and the *technology viewpoint* (the specific technologies chosen for the implementation). Figure 1 shows the five RM ODP viewpoints and their dependencies, with the focus in this article being on the computational viewpoint.

Earlier research work by the Commission began with a general overview of the SDI (Aalders & Moellering 2001), and then proceeded to examine SDIs using UML (Cooper *et al* 2003). Subsequently, the Commission developed formal models of SDIs from the enterprise and information viewpoints of RM-ODP (Hjelmager *et al* 2005; Hjelmager *et al* 2008). The investigation of the enterprise viewpoint of an SDI identified the stakeholders of an SDI and their activities, i.e. the actors and the use cases, in UML; and identified the core components of an SDI and their relations.

While the enterprise viewpoint deals mainly with the administrative aspects of an SDI, the information viewpoint has its focus on the products (data and services), their specification, their description via metadata, and product registries (catalogues). For the information viewpoint, we identified *information classes* that are required for

SDIs to deliver data and services and showed how stakeholders are linked to the information classes through *stakeholder activities* in an SDI.

2.2 Other attempts at modelling SDIs

There are a number of documents that publish guidelines or specifications for the software architectures of specific SDI implementations. For example, the INSPIRE Network Services Architecture (INSPIRE Network Services Drafting Team 2008) describes the INSPIRE service types that are mandated by the INSPIRE Directive (2007) and shows how these services connect to portals and applications through the INSPIRE service bus. Another example is the Canadian Geospatial Data Infrastructure (CGDI) (Geoconnections 2005), which provides a high-level overview of the CGDI architecture and describes the range of services that comprise the CGDI. The CGDI architecture endorses the RM-ODP, and describes elements that can be used to support the information, engineering and computational RM-ODP viewpoints, while the implementation details of the RM-ODP technology and engineering viewpoints are left as the responsibility of agencies collaborating in the CGDI. Both the INSPIRE Network Service Architecture and the CGDI architecture are technology independent, similar to the Commission's scientific model, but both of them are bound by the policies and legislation of the respective organizations. The Commission's model recognizes that policies and legislation play a role in an SDI but the model is independent of specific policies and legislation, an approach also taken for the SDI Cookbook (Nebert 2004).

Béjar *et al* (2008) analyze published architectural SDI models and based on this analysis, propose an architectural style for SDI software architectures. This style provides a tool and a shared vocabulary to help system architects to design SDIs, and facilitates the exchange of knowledge about them. The style is defined under the component-and-connector architectural viewtype, extending the client-server and shared-data styles. In contrast, the functional decomposition into a set of objects provided in the computational viewpoint of the Commission's SDI model provides the basis for decisions on how to distribute objects. The Commission's model is therefore not restricted to client-server architectures, but can also be implemented in emerging architectures, such as data grids and cloud environments. In addition, the Commission's work provides a holistic view of an SDI by linking components of the different viewpoints, for example, by listing SDI services that are performed by objects (computational viewpoint) during stakeholder (enterprise viewpoint) activities associated with information classes (information viewpoint).

Coetzee (2009) presents Compartimos, a reference model for sharing address data on a data grid in an SDI environment. The computational viewpoint for Compartimos describes the essential components, and their relationships and interactions, that are required for sharing address data on a data grid. In Coetzee and Bishop (2009), commonly available technology choices for Compartimos are explored. Compartimos is, similar to the extended client-server and shared-data architectural styles of Béjar *et al* (2008) because it is specific in terms of an architectural style, namely the data grid. The Commission's model from a computational viewpoint, on the other hand, does not specify the physical distribution of the computational objects and is therefore not bound to a specific architectural style. The actual distribution of computational objects could be specified in future in one or more engineering viewpoints. In addition, while Compartimos is specific to address data, the Commission's model applies to any kind of spatial information.

While there are those who have nominally applied systems theory to SDIs (e.g. Mavima *et al* (2001)), they have not actually done so (ie: considering SDIs as self-regulating systems), but have rather applied various theories of systems to SDIs (soft systems theory, in the case of Mavima *et al* (2001)). Grus *et al* (2006) considered SDIs as complex adaptive systems, which are systems that changed from stable and predictable to unstable and unpredictable, and then exhibited more complex patterns of behaviour. Mansourian & Abdolmajidi (2011) applied system dynamics to SDIs, that is, where positive and negative feedback loops are used to model systems with multiple components that mutually interact within a common goal and where the system adjusts internally in response to external disturbances.

3. Description of an SDI from a computational viewpoint

The *computational viewpoint* is a functional decomposition of the modelled system into a *set of objects* that interact at *interfaces*, such as to provide services – enabling system distribution (ISO/IEC 10746-1:1998). In the context of an SDI, the computational viewpoint captures the details of the computational objects and their interfaces definitions without regard to the physical distribution of the objects (i.e. where they actually are located in the world). The latter is covered by the engineering viewpoint. The computational viewpoint sets the scene for distribution by decomposing the system and by specifying a binding model describing how interactions between given computational interfaces are carried out.

3.1 SDI computational objects modelled as UML components

We use a UML component diagram (Object Management Group 2010) to model the SDI computational objects. The component diagram allows one to structure the *components* of a system (autonomous and encapsulated units within the system providing one or more interfaces) and their interrelations. Briefly, the basic elements of a component diagram are (as shown in Figure 2):

- *Component* (rectangle with small symbol in upper right corner): A *component* represents a modular part of a system that encapsulates its contents and whose manifestation is replaceable within its environment. A component defines its behavior in terms of provided and required interfaces.
- *Provided Interface* (connector with circlet): An interface is a named set of operations that characterize the behaviour of a component. A *provided interface* is one that is implemented directly by the component.
- *Required Interface* (connector with arc): A *required interface* specifies services that a component needs in order to perform its function and fulfill its own obligations to its clients.
- *Dependence* (dashed arrow): A *dependency* is a relationship that signifies that a component requires services from an other component through its *provided interface* (Object Management Group 2010).

The RM-ODP defines a computational interface as being characterized by a signature, a behaviour and an environmental contract. However, since we are modelling the SDI computational objects as UML components, we specify the interfaces of the provided and required interfaces of the SDI computational objects, as defined in the list above. For simplicity reasons, we did not model the details of the signature and contract of individual interfaces. These could be provided in a further refinement of the computational viewpoint.

Figure 2 shows the six SDI computational objects in a UML component diagram: *SDI Registry*, *SDI Data*, *SDI Processing*, *SDI Application*, *SDI Portrayal* and *SDI Management*. Each SDI computational object *offers* a number of functionalities, modelled by the *provided interfaces*, as well as *uses* functionality offered by other SDI computational objects, modelled by the *required interfaces*.

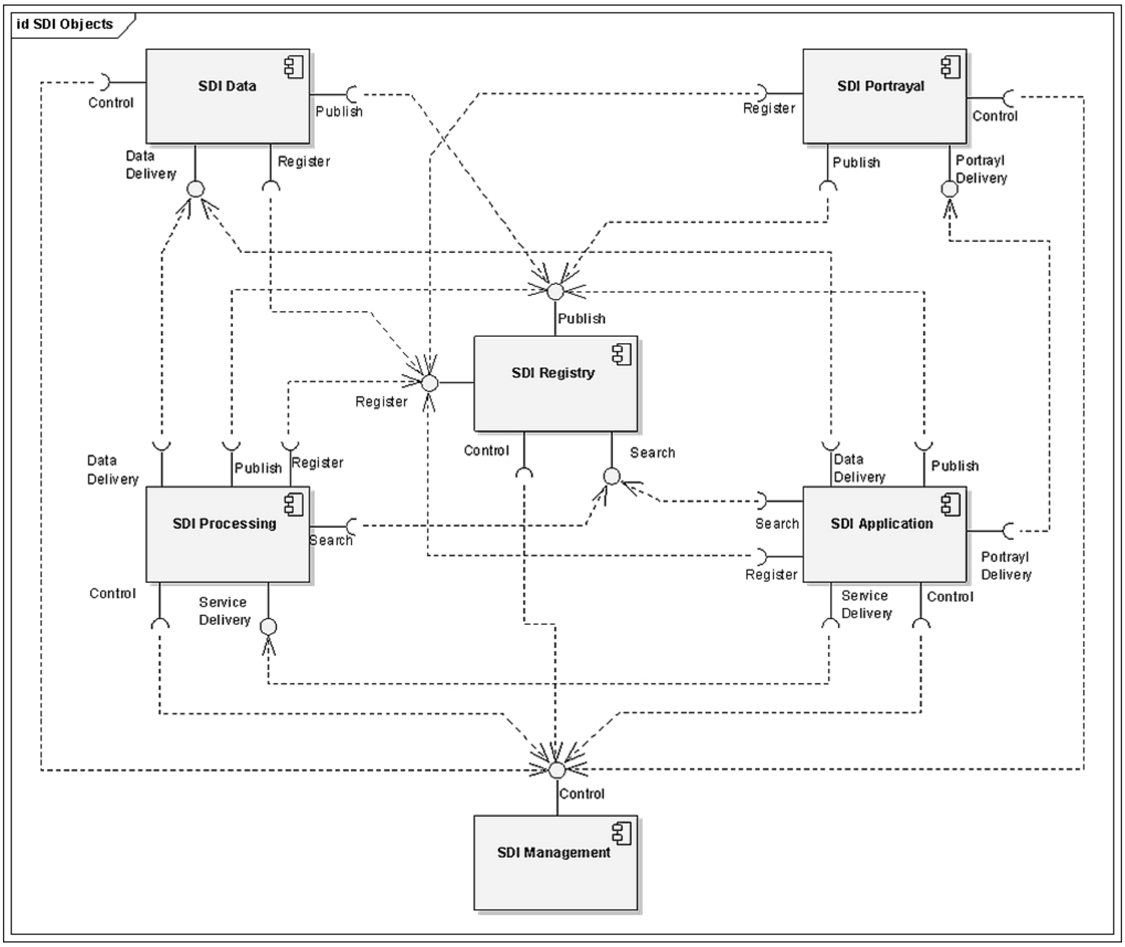

Figure 2. The SDI computational objects (adapted from Cooper *et al* 2007)

3.2 Purpose and interfaces of the SDI computational objects

3.2.1 SDI Registry

The main purpose of the *SDI Registry* is to *register* data, services (i.e. products) and other items in the catalogue, to *publish* them, and afterwards let users *search* through them. This functionality is provided by the three interfaces of *SDI Registry* described below. The *SDIManagement::Control* is the only required interface. The double colon notation '::' means Interface.

The *SDIRegistry::Register* interface provides the necessary operations to *register* information about resources available on a network, such as, Register Product Specification, Register Product, Register Metadata, Register Catalogue, Register Policy, Register Business Plan and Update Register

The *SDIRegistry::Search* interface facilitates searching for the required items in the relevant catalogue and includes operations, such as, Search Register, Search Product Specification, Search Product, Search Metadata, Search Catalogue and Search through Catalogue.

The *SDIRegistry::Publish* interface provides functionality for publishing the output information from Registers through the Internet or other media. Operations, such as, Publish Product Specification, Publish Product, Publish Metadata and Publish Catalogue are typically included.

3.2.2 SDI Data

SDI Data deals with data sets shared and registered on the Internet. For example, *SDI Data* provides access to collections of data in repositories and databases. The only provided interface is *SDIData::DataDelivery*, which is designed in such a way that it provides data to users via the *SDI Processing* or *SDI Application*. No other SDI computational object deals with data sets directly. Required interfaces are *SDIRegistry::Register*, *SDIRegistry::Publish* and *SDIManagement::Control.*

3.2.3 SDI Processing

SDI Processing provides the interfaces for data processing, such as, coordinate computation and projection system transformation. It provides only one interface, *SDIProcessing::ServiceDelivery*. However, a number of interfaces are required, including *SDIRegistry::Register*, *SDIRegistry::Publish*, *SDIRegistry::Search*, *SDIData::DataDelivery* and *SDIManagement::Control.*

3.2.4 SDI Application

SDI Application is a key part of the SDI computational architecture. It does not provide any interface, but requires a large number of interfaces in order to meet the needs of users. These include SDIRegistry::Register, *SDIRegistry::Publish*, *SDIRegistry::Search*, *SDIData::DataDelivery*, *SDIProcessing::ServiceDelivery*, *SDIPortrayal:: PortrayalDelivery* and *SDIManagement::Control.*

3.2.5 SDI Portrayal

SDI Portrayal deals with displaying the results of application services. It provides one interface for these purposes, *SDIPortrayal::PortrayalDelivery*, which facilitates output, such as, designing layouts, editing functions, specifying delivery options and formats. *SDI Portrayal* has the following required interfaces: *SDIRegistry::Register*, *SDIRegistry::Publish* and *SDIManagement::Control.*

3.2.6 SDI Management

SDI Management monitors the overall functionality of the SDI. For this purpose it has one interface, *SDIManagement::Control,* which, for example, controls the interoperability amongst services or rights of access.

3.3 Binding model

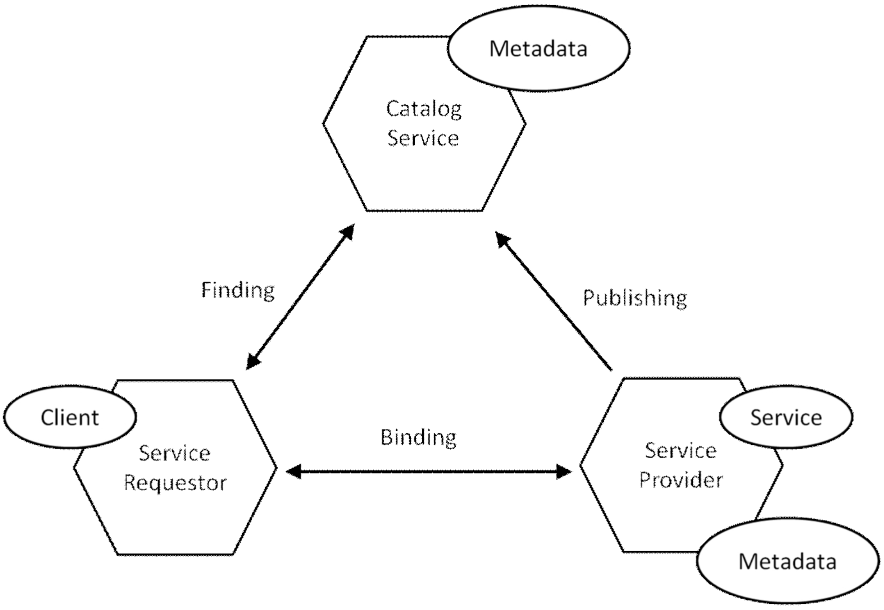

Figure 3. A service oriented architecture (adapted from W3C 2004)

According to the RM-ODP (ISO/IEC 10746-1:1995), binding should be modelled in a computational viewpoint. The purpose of the binding model is to specify how liaisons are to be created between objects in order for interactions to occur.

The ICA computational model of an SDI is a service-oriented architecture (SOA), as shown in Figure 3 (W3C 2011). An SOA is based on the concept that a service provider publishes its services at a service registry. A service requester finds details about a specific service at the registry, proceeds to bind to the service at the service provider and starts interacting with the service at the service provider. Since our model is technology independent, the binding model is inherited from whichever technology is used in the implementation (e.g. SOAP, REST, DCOM or CORBA).

4 SDI services provided by SDI computational objects for stakeholder activities

In section 3, we identified the *provided* and *required* interfaces of SDI computational objects. In this section we show how stakeholders (identified in the enterprise viewpoint) interact through these interfaces during the activities associated with the information classes (identified in the information viewpoint).

Initially, the Commission defined the concept "service" for its own purpose, starting with the definitions from *ISO 19119:2005, Geographic information – Service*. As such, an *SDI service* was defined as *the distinct part of the SDI functionality that is provided by an SDI computational object through interfaces, either manually or automatically.*

Additionally, an *interface* is defined as *a named set of operations that characterize the behaviour of an object.*

Moreover, an *operation* is defined as *something that an object may be called on to do, may do without request, or may do for internal reasons.*

By the integration of these definitions, an *SDI service* is defined as *the distinct part of the SDI functionality that is provided by an SDI computational object through named sets of operations that characterize the behaviour of the SDI computational object, where each operation may be called on to do, may do without request, or may do for internal reasons, either manually or automatically.*

Thus, in this model, an *SDI service* is the distinct part of the SDI functionality that is provided by a collection of operations from (one or more) interfaces of (one or more) SDI computational objects. An operation is part of a single interface and is performed by a single SDI computational object, but a service is more general, collectively referring to a number of operations from one or more computational object's interfaces.

Based on this definition of an SDI service, we identified more than 180 services within an SDI and classified them in a matrix showing which computational object provides them and where they are required for stakeholder activities associated with the five information classes (i.e. Policies, Product specification, Product, Metadata, and Catalogue) from the information viewpoint. Table 1 provides a detailed list of the SDI services. Column 1 contains the information classes defined in the information viewpoint (from Table 1 in Hjelmager *et al* 2008). Column 2 contains the stakeholder activities associated with these classes, also defined in the information viewpoint (from Table 1 in Hjelmager *et al* 2008). Row 1 contains the SDI computational objects. The rest of the cells in the table show the SDI services that fulfil the functionality required by the stakeholder activities. Note that there does not have to be a specific SDI service linking every activity to every object in the table.

Consider, for example, the *Publishing* service, which makes all kinds of information publicly available. The *SDI Registry* computational object provides the required interfaces. The *Publishing* service is required in all the information classes for various stakeholder activities, such as, ‘Make policy’ and ‘Make business plan’ of the Policies information class and ‘Stipulate requirements’, ‘Translate into product specifications’ and ‘Obtain and implement product specifications’ of the Product specifications information class.

Another example is the *Delivery* service, which delivers data, metadata and catalogues to SDI stakeholders. *SDI Data* and *SDI Processing* provide the *SDIData::Delivery* and *SDIProcessing::ServiceDelivery* interfaces for the ‘Provide product’, ‘Use product’ and ‘Maintain product’ stakeholder activities of the Product information class; the ‘Provide metadata’ of the Metadata information class and the ‘Provide catalogue’ of the Catalogue information class. *SDI Application* does not provide interfaces but acts as intermediary to deliver data, metadata and catalogues through the above mentioned *SDI Data* and *SDI Processing* interfaces to the SDI stakeholders in the ‘Provide product’ and ‘Provide catalogue’ of the Product and Catalogue information classes respectively.

Table 1: SDI services provided by computational objects (columns) for stakeholder activities in each of the SDI information classes (rows)

Information Class	**SDI computational object** / **Activity**	**SDI Registry**	**SDI Data**	**SDI Processing**	**SDI Portrayal**	**SDI Application**	**SDI Management**
Policies	Make policy	Publishing	-	-	-	-	Development Updating
	Apply policy	Implementation Registration Searching	-	-	-	-	Adherence Monitoring
	Make business plan	Registration Publishing	-	-	-	-	Development Implementation Updating
	Use business plan	Implementation Searching	-	-	-	-	Adherence Monitoring Implementation
Product specifications	Consult users	Registration	-	-	-	-	Monitoring Survey Relationship Management
	Stipulate requirements	Registration Publishing	-	-	-	-	Development
	Translate into product specification	Registration Publishing	-	-	-	-	Development
	Obtain and implement product specification	Publishing Searching	Implementation	Implementation	Implementation	Implementation	Certification Accreditation

Information Class	SDI computational object Activity	SDI Registry	SDI Data	SDI Processing	SDI Portrayal	SDI Application	SDI Management
Product	Capture/create data (from source)	Registration Publishing	Storing	Sensor Access Transformation	-	Capture	-
	Produce product	Registration Publishing	Storing	Transformation	Design	Transformation Application Chaining	-
	Assure quality (production process)	Registration Publishing	-	Calibration Reporting	-	-	Auditing
	Assure quality (certification of product)	Registration Publishing	Sampling Extraction Storing	Quality Evaluation Quality Assurance Reporting	Representation	Quality Evaluation Quality Assurance Reporting	Auditing Certification
	Provide product	Searching	Extraction Delivery Access	Delivery Access	Representation	Delivery Access	Adherence Monitoring
	Use product	Searching	Extraction Delivery Access Assessment	Transformation	Design Representation	Application Chaining Assessment	Adherence Monitoring
	Maintain product	Updating	Extraction Storing Delivery Access Updating	Transformation Updating	Design Updating	Transformation Application Updating Chaining	Scheduling
Metadata (incl. Service	Produce metadata	Registration Publishing	Assessment	Capture	-	Capture Service	Monitoring

Information Class	**SDI computational object** **Activity**	**SDI Registry**	**SDI Data**	**SDI Processing**	**SDI Portrayal**	**SDI Application**	**SDI Management**
capability)	Assure quality of metadata	Registration Publishing Sampling Extraction Updating	Sampling Extraction	-	Representation	Evaluation Assurance Reporting	Auditing
	Provide metadata	Searching Extraction	-	Delivery	Representation	-	Adherence Monitoring
	Harvest metadata	Searching Extraction Updating Registration Publishing	-	Harvesting Transformation	-	Assessment Transformation Chaining	Monitoring Scheduling
	Search through metadata	Searching	-	-	Representation	Chaining	Monitoring
	Analyse metadata	-	-	Transformation	Representation	Assessment Transformation Chaining	-
	Maintain metadata	Updating	Assessment Service	Capture	-	Capture	Monitoring Scheduling
Catalogue	Produce catalogue	Registration Publishing Classification	-	Capture	-	Capture	Monitoring
	Provide catalogue	Searching Extraction	-	Delivery	Representation	Delivery	Monitoring
	Search for catalogue (incl. chaining)	Searching	-	-	Representation	Chaining	Monitoring
	Search through catalogue	Searching	-	-	Representation	Chaining	Monitoring
	Maintain catalogue	Updating	-	Capture	-	Capture	Monitoring Scheduling

5. SDI services for SDI development and use

The development and use of an SDI requires eight processes (Cooper *et al* 2009) and these are illustrated in Figure 4. The Liaison process is not shown as an arrow because it links together everything. The roles of the processes can be described in this way:

- ***Initiation***: includes the initial idea for the SDI, assembling the relevant stakeholders, developing the framework for the SDI and motivating it.
- ***Creation***: establishing the SDI.
- ***Management***: ongoing management of the SDI.
- ***Manipulation***: includes finding and using metadata and finding data.
- ***Access***: covers accessing data, including production of customised real and virtual maps.
- ***Processing***: includes transforming, integrating and modelling data.
- ***Evaluation***: assessing the functioning of the SDI and the quality and utility of the data, metadata and services offered by the SDI.
- ***Liaison***: maintaining relationships with stakeholders.

These processes represent different ways of interacting with an SDI and in this regard, have different needs for SDI services provided by the computational objects. We describe the processes in subsequent subsections. Some examples from INSPIRE are included to illustrate how computational objects provide the required services for these processes.

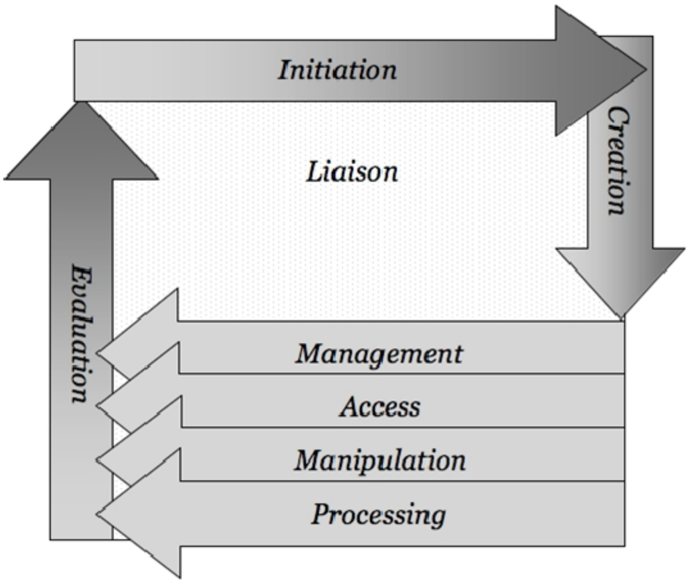

Figure 4. The eight SDI processes

Table 2 provides an overview of SDI services required by different SDI processes. The rows represent SDI processes and the columns represent computational objects. In the intersection of a row and a column are listed all the relevant SDI services provided by the given computational object (of that table column) in the given SDI process (of that table row).

5.1. Initiation

During *Initiation*, the strategy, concepts and rules for the proposed SDI are developed. *SDI Management* plays an important role in this process: it should provide the necessary *Development* service. Further, all computational objects should afterwards implement this strategy, this concept and these rules through the provided *Implementation* services in this process.

For example, in INSPIRE the European Union represents some aspects of SDI Management. It provides a Development service by creating policies, such as the INSPIRE Directive of 2007. Other aspects of SDI Management are represented by the relevant authorities in the Member States who set up business plans for INSPIRE activities.

5.2. Creation

The main goal of this process is to create the SDI. The SDI services provided by computational objects in this process are much richer and differ substantially from one object to another, as described in Sections 3 and 4.

SDI Registry should start to register *Products*, *Metadata*, *Registers*, etc. *SDI Data* provides services focused on *Storing* data sets. The *SDI Processing* starts *Harvesting*, *Capturing*, and *Reporting* on metadata. *SDI Portrayal* provides a *Design* service to support the production and use of products. *SDI Application* supports *Capture* and *Reporting* functionalities.

For example, the INSPIRE Geoportal (http://inspire-geoportal.ec.europa.eu/editor/) resembles the *SDI Application* computational object. The Geoportal's Metadata Editor provides a *Capture* service: it allows users to create metadata according to the INSPIRE implementing rules. The Geoportal's Validator provides a Reporting service: it tests compliance of INSPIRE metadata against the INSPIRE Metadata Regulation.

5.3. Management

The management of an SDI service is, as was the case with creation, much richer and differs substantially from one computational object to another, as described in sections 3 and 4.

Table 2: SDI services provided by computational objects (columns) and required by the different SDI processes (rows)

SDI computational object **Process in an SDI**	**SDI Registry**	**SDI Data**	**SDI Processing**	**SDI Portrayal**	**SDI Application**	**SDI Management**
Initiation	*Implementation*	*Implementation*	*Implementation*	*Implementation*	*Implementation*	*Development* *Implementation*
Creation	*Registration*	*Storing*	*Capture* *Reporting* *Harvesting*	*Design*	*Capture* *Reporting*	
Management	*Registration* *Updating*	*Storing* *Updating*	*Capture* *Reporting* *Updating* *Harvesting*	*Design* *Updating*	*Capture* *Reporting* *Updating*	*Updating* *Scheduling*
Manipulation	*Searching* *Sampling* *Extraction* *Publish*	*Sampling* *Extraction* *Delivery*	*Delivery*		*Delivery*	
Access	*Access*	*Access*	*Access*	*Representation*	*Access*	*Adherence*
Processing			*Transformation*		*Transformation* *Application* *Chaining*	
Evaluation	*Classify*	*Assessment*	*Calibration* *Assessment*		*Assessment*	*Monitoring* *Survey* *Assurance* *Auditing*
Liaison	*Stakeholder tracking*					*Relationship Management*

SDI Management should provide a *Scheduling* service. *SDI Registry* should manage registration of Products, Metadata, Registers, etc. *SDI Data* provides services focused on manage stored data sets. *SDI Processing* manages *Harvesting*, *Capturing*, and *Reporting* on metadata. *SDI Portrayal* provides a Design service to support and manage the production and use of products. *SDI Application* supports the management of the *Capture* and *Reporting* functionalities. All computational objects are part of the *Update* service and thereby help in the management of an SDI.

In INSPIRE the Member States and the INSPIRE Geoportal represent *SDI Data* in the *Updating* and *Storing* services respectively when data is updated and subsequently made available in the INSPIRE Geoportal. The European Commission Decision 2009/442/EC of 5 June 2009 states that Member States have to report annually a number of indicators for monitoring the implementation and use of their infrastructures for spatial information. Various other Commission regulations set deadlines for regulation implementations by Member States. Here the Commission represents *SDI Management* in the *Scheduling* service.

5.4. Manipulation

The Manipulation process bears on the use of the SDI by users. That is why its activities (and the services provided) concentrate mainly on the *SDI Registry*, *SDI Data* and *SDI Processing* computational objects.

SDI Registry supports the *Searching*, *Sampling*, *Extraction* and *Publishing* of *Metadata* and *Products* while *SDI Data* supports *Sampling*, *Extraction* and, primarily, *Delivery* to enable the use of data. *SDI Processing* and *SDI Application* play important roles as manipulation services – they support the *Delivery* service (besides the manipulative role) to deliver *Metadata*, *Catalogues*, and *Products* to users.

The INSPIRE Geoportal represents *SDI Registry* because it supports the *Searching* and *Publishing* service. It also represents the *Delivery* service by making the data available in a Web browser, i.e. a *Delivery* service.

5.5. Access

The Access process is supported by the *Access* service allowing access to the *Registry* and *Product* while controlled by the *Registry* service. The *Registry* service controls who and to what part of the data the user in question is allowed access. These services are provided by the *SDI Data*, *SDI Processing*, and *SDI Application* computational objects. *SDI Portrayal* provides the *Representation* service and *SDI Management* provides the *Adherence* service.

Once again the INSPIRE Geoportal represents more than one computational object, including *SDI Data*, *SDI Processing* and *SDI Application,* when providing the *Access* service to data and services.

5.6. Processing

This process is supported mainly by *SDI Processing* and *SDI Application*. Both provide a *Transformation* service, and the latter one also provides *Application* and *Chaining* services, as processing will invariably consist of multiple services chained together.

At the moment it does not seem as if the INSPIRE Geoportal provides *Transformation* and *Chaining* services, but it could well do that in the future.

5.7. Evaluation

SDI Evaluation provides the widest suite of services in this process, taking care of the *Monitoring*, *Survey*, *Assurance*, and *Auditing* services. *SDI Data*, *SDI Processing*, and *SDI Application* provide the *Assessment* service. Additionally, SDI *Register* supports the *Classification* service, and *SDI Processing* provides the *Calibration* service.

Here the European Commission through its Decision 2009/442/EC of 5 June 2009 represents *SDI Evaluation* in the *Monitoring* service.

5.8. Liaison

SDI Management and *SDI Registry* act in this liaison process. The former provides the *Relationship Management* service and the latter the *Stakeholder Tracking* service.

Information about registered INSPIRE stakeholders is available on the INSPIRE website, representing SDI *Registry* and providing the *Stakeholder Tracking* service.

6. Conclusions

In this article, a novel approach has been used to model SDIs based on the RM-ODP viewpoints. This approach contributes to a better understanding of the concept of an SDI. This article presents the recent contribution of the ICA Geoinformation Infrastructures and Standards Commission's model towards a holistic understanding of an SDI that is independent of specific SDI legislation, technology, and implementation. The work presented in this article is in the continuity of the previous Commission work on the enterprise and information viewpoints of RM-ODP (Hjelmager *et al* 2005, Hjelmager *et al* 2008).

This article has focused on the description of an SDI model from the RM-ODP computational viewpoint. The proposed computational view point model is composed of six computational objects, i.e. *SDI Registry*, *SDI Data*, *SDI Processing*, *SDI Application*, *SDI Portrayal*, and *SDI Management*. Each computational object offers functionalities as well as requests and uses functionalities from other objects. These functionalities are modelled by the way of interfaces in a UML component diagram.

The types of functionalities are identified for each computational object along with their dependencies. This proposed SDI model from a computational viewpoint simplifies the description of an SDI by representing related functionality in a single computational object.

Additionally, a definition of SDI service has been proposed and, based on it, required services have been classified by information class (i.e. policies, product specifications, product, metadata, and catalogue), stakeholder activities, and computational objects in a matrix (Table 1). Examples of the use of the classification matrix have been given.

Finally, this article has analyzed and identified the requirements for SDI services from the identified computational objects based on the processes for the development and use of an SDI (i.e. initiation, creation, management manipulation, access, processing, evaluation, and liaison). The SDI services have been illustrated with examples from the INSPIRE implementation of an SDI. For example, SDI Management in the INSPIRE implementation corresponds to various computational objects such as registration, storing, updating and scheduling services, and the INSPIRE Geoportal implements more than one computational object depending on the service that it provides (see the description of the Creation and Manipulation processes in the previous section). Our proposed computational model makes it possible to describe an SDI independently of its implementation.

In the current ICA computational viewpoint SDI model, it has been chosen for simplicity reasons to exclude the details (e.g. signature and contract) of individual interfaces. However, future work will provide a further refinement of the computational viewpoint. Also as part of future work, the assessment of the proposed SDI model from the different viewpoints (enterprise, information and computational viewpoints) must be considered by applying it to specific SDI communities. This will allow one to identify weaknesses and to propose refinements. No attempts to model an SDI from the engineering or technology viewpoints were considered yet, since these are meant to be implementation-specific, whereas the Commission is oriented towards providing technology independent models for an SDI. However, the reverse engineering of the engineering and technology viewpoints of an existing SDI implementation could be considered to provide additional feedback and benefits to both the computational viewpoint and the SDI implementation.

7. Acknowledgements

This research has been partially supported by various organizations including:

- The Ohio State University has provided in-kind support and access to electronic communication systems, and office space for the ICA Commission on Spatial Data Standards since 1989.
- This work was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070).

- The South African National Research Foundation (NRF).
- The Centre for SDIs and Land Administration, Department of Geomatics, The University of Melbourne and Mapping Science Institute, Australia (MSIA).
- Iberian-American Project CYTED IDEDES 606AC0294 “Evaluation and strengthening of Spatial Data Infrastructure for sustainable development for Latin-American and the Caribbean”.
- The South Africa/Poland Agreement on Cooperation in Science and Technology fort the Joint Research Grant, "Volunteered Geographical Information (VGI) for Spatial Data Infrastructures (SDIs) and Geoportals".
- The Polish Ministry of Science and Higher Education for the project “Budowa, standaryzacja i integracja sieciowych usług geoinformatycznych dla potrzeb opracowań kartograficznych”, number N N526 193936.

We would also like to acknowledge the excellent support we received from Prof German Kuroshev of the Saint-Petersburg State University, who hosted our Commission meeting in 2007. Finally, we wish to dedicate this to our dear colleague, Alexander Martynenko, who passed away in September 2007.

References

Aalders H.J.G.L. & Moellering, H., 2001. Spatial Data infrastructure, *Proceedings of the 20th International Cartographic Conference, Beijing, China*, Vol. 4, pp. 2234-2244.

Béjar R., Latre M.A., Nogueras-Iso J., Muro-Medrano P.R. and Zaragoza-Soria F.J., 2009. An architectural style for spatial data infrastructures, *International Journal of Geographical Information Science*, 23(3-4), pp 271–294.

Coetzee S. and Bishop J., 2009. An analysis of technology choices for data grids in a spatial data infrastructure. *In:* B. van Loenen, J.W.J. Zevenbergen and J.A Besemer eds. *Spatial Data Infrastructure Convergence: Research, Emerging Trends, and Critical Assessment.* Delft, Netherlands: Nederlandse Commissie voor Geodesie/Netherlands Geodetic Commission, 48, pp107-120.

Coetzee S. 2009. *An analysis of a data grid approach for spatial data infrastructure*. PhD Computer Science, University of Pretoria. Available from: http://upetd.up.ac.za/thesis/available/etd-09272009-152926/ [Accessed 12 October 2012].

Cooper A.K., Hjelmager J., Nielsen A. and Rapant P., 2003. A Description of Spatial Data Infrastructures (SDIs) Using the Unified Modelling Language (UML). *Proceedings of the 21st International Cartographic Conference*, August 2003, Durban, South Africa. Available from: http://researchspace.csir.co.za/dspace/handle/10204/1776 [Accessed 12 October 2012].

Cooper A.K., Moellering H., Delgado T., Düren U., Hjelmager J., Huet M., Rapant P., Rajabifard A., Laurent D., Iwaniak A., Abad P., & Martynenko A., 2007. An initial model of the computation viewpoint for a spatial data infrastructure. *23rd International Cartographic Conference,* August 2007, Moscow, Russia.

Available from: http://researchspace.csir.co.za/dspace/handle/10204/1455 [Accessed 12 October 2012].

Cooper A.K., Moellering H., Hjelmager J., Rapant P., Laurent D., Abad P. and Danko D., 2009. Detailed Services in a Spatial Data Infrastructure from the Computation Viewpoint. *Proceedings of the 24th International Cartographic Conference*, November 2009, Santiago, Chile. Available from: http://researchspace.csir.co.za/dspace/handle/10204/3822 [Accessed 12 October 2012].

Delgado T. 2005. *Infraestructuras de Datos Espaciales en países de bajo desarrollo económico. Implementación en Cuba.* Tesis de doctorado, ITM "José Martí".

GeoConnections, 2005. *The Canadian Geospatial Data Infrastructure: Architecture Description Version 2.0*. Available from: ftp://ftp2.cits.rncan.gc.ca/pub/geott/ess_pubs/288/288844/cgdi_ip_03_e.pdf [Accessed 12 October 2012].

Groot R. and McLaughlin J., 2000. *Geospatial Data Infrastructure: Concepts, Cases and Good Practices.* Oxford: Oxford University Press. ISBN 0198233817.

Grus L., Crompvoets J. & Bregt A.K., 2010. Spatial data infrastructures as complex adaptive systems. *International Journal of Geographical Information Science*, 24(3), pp439-463.

Hjelmager J., Delgado T., Moellering H., Cooper A.K., Danko D., Huet M., Aalders H.J.G.L. & Martynenko A., 2005. Developing a modelling for the spatial data infrastructure. *Proceedings of the 22nd International Cartographic Conference*, July 2005, A Coruña, Spain. Available from: http://researchspace.csir.co.za/dspace/handle/10204/1781 [Accessed 12 October 2012].

Hjelmager J., Moellering H., Delgado T., Cooper A.K., Rajabifard A., Rapant P., Danko D., Huet M., Laurent D., Aalders H.J.G.L., Iwaniak A., Abad P., Düren U. and Martynenko A., 2008. An initial Formal Model for Spatial Data Infrastructures. *International Journal of Geographical Information Science*, 22 (11), pp1295 - 1309.

INSPIRE Network Services Drafting Team, 2008. *Network Services Architecture*. Available from: http://inspire.jrc.ec.europa.eu/reports/ImplementingRules/network/D3_5_INSPIRE_NS_Architecture_v3-0.pdf [Accessed 12 October 2012].

Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE) [2007] OJ l 108/1. Available from: http://inspire.jrc.ec.europa.eu/ [Accessed 12 October 2012]

ISO/IEC 10746-1:1998, *Information Technology --- Open Distributed Processing --- Reference Model: Overview*, International Organization for Standardization (ISO), Geneva, Switzerland.

ISO/IEC 10746-2:1996, *Information Technology --- Open Distributed Processing --- Reference Model: Foundations*, International Organization for Standardization (ISO), Geneva, Switzerland.

ISO/IEC 10746-3:1996, *Information Technology --- Open Distributed Processing --- Reference Model: Architecture*, International Organization for Standardization (ISO), Geneva, Switzerland.

ISO/IEC 10746-4:1998, *Information Technology --- Open Distributed Processing --- Reference Model: Architectural Semantics*, International Organization for Standardization (ISO), Geneva, Switzerland.

ISO 19119:2005, *Geographic information – Services*, International Organization for Standardization (ISO), Geneva, Switzerland.

ISO/IEC 19501:2005*, Information Technology --- Open Distributed Processing --- Unified Modeling Language (UML) Version 1.4.2*, International Organization for Standardization (ISO), Geneva, Switzerland.

Mansourian A. & Abdolmajidi E., 2011. Investigating the system dynamics technique for the modeling and simulation of the development of spatial data infrastructures. *International Journal of Geographical Information Science*, 25(12), pp2001-2023.

Masser I, 2005. *GIS Worlds - Creating Spatial Data Infrastructures*. ESRI Press, ISBN 1589481224.

Mavima R., Barry M. & Rivett U., 2001. Systems Thinking: An Approach Towards the Development of Spatial Data Infrastructures: The Case of Zimbabwe. *FIG International Conference on Spatial Information for Sustainable Development*, Nairobi, Kenya. Available from: http://www.fig.net/pub/proceedings/nairobi/mavima-barry-rivett-TS2-4.pdf [Accessed 12 October 212].

Moellering H. (ed.), 2000. Special Issue on the Nature of Analytical Cartography, *Cartography and Geographic Information Science*, 27(3), pp. 187-254.

Nebert D.D., 2004. *Developing spatial data infrastructures: The SDI Cookbook*. Available from: http://www.gsdi.org/docs2004/Cookbook/cookbookV2.0.pdf [Accessed 1 September 2012]

Object Management Group, 2011. *OMG Unified Modeling Language™ (OMG UML) version 2.4.1*. Available from: http://www.omg.org/spec/UML/2.4.1/ [Accessed 12 October 2012].

Rajabifard A., Binns A., Masser I. & Williamson I.P., 2006. The Role of Sub-national Government and the Private Sector in Future SDIs. *International Journal of Geographical Information Science*, 20(7), pp727-741.

W3C, 2004. *Web Services Architecture*. Available from: http://www.w3.org/TR/ws-arch/ [Accessed 12 October 2012].

W3C, 2011. *World Wide Web Consortium*. Available from: http://w3.org/ [Accessed 12 October 2012].