

An RM-ODP Enterprise View for Spatial Data Infrastructures

Rubén Béjar^a, Miguel Á. Latre, Javier Nogueras-Iso, Pedro R. Muro-Medrano, F. Javier Zarazaga-Soria

Computer Science and Systems Engineering Department, Universidad de Zaragoza, c/María de Luna 1, 50018, Zaragoza, Spain

^a*rbejar@unizar.es*

Abstract

Spatial Data Infrastructures (SDIs) are large, open, distributed and standards-based information systems which intend to facilitate and promote the use of spatial data and spatial services on the Internet. Spatial data describe information tied with locations on Earth, while spatial services allow to manipulate spatial data following a Service Oriented Architecture. This paper proposes to model SDIs as federations of autonomous communities following the enterprise language of the ITU-T and ISO/IEC ‘Reference Model of Open Distributed Processing’ (RM-ODP), and the recently approved ‘Use of UML for ODP Systems Specifications’ (UML4ODP). The enterprise language of the RM-ODP provides a conceptual foundation to address several aspects of SDIs not previously considered from a systems architecture point of view. The use of UML4ODP provides a modelling language to facilitate the exchange of knowledge about SDI, and it is an opportunity to try this recent standard for a class of large and complex systems.

Keywords: RM-ODP, Enterprise Architecture, Service Oriented Architecture, Spatial Data Infrastructure, UML, Distributed System

1. Introduction

The importance of spatial data to support decision-making and management has been cited as critical in important United Nations (UN) events such as the 1992 Rio Summit, the special session of the UN General Assembly to

appraise the implementation of the Agenda 21 in 1997, or the World Summit on Sustainable Development in Johannesburg in 2003 [1]. Spatial, or geographic, data describe information tied to some locations on Earth's surface or to zones adjacent to it.

In the last decade of the 20th century, the use of spatial data was limited due to its high prices and the use of closed and monolithic systems. To improve that situation, the development of Spatial Data Infrastructures (SDIs) was proposed as a means to facilitate the discovery, access and use of spatial information. According to the 'SDI cookbook', 'the term 'Spatial Data Infrastructure' (SDI) is often used to denote the relevant base collection of technologies, policies and institutional arrangements that facilitate the availability of and access to spatial data' [1, p. 8]. Other relevant definitions for the term SDI are cited in [2], but in general these definitions share common objectives and similar components, which have been similarly categorized by different authors: from the people, policies and agreements, standards and technologies proposed in [3] to the framework composed of data, people, institutional frameworks, technology and standards in [4, p. 22-23]. SDIs are complex systems, and under this point of view have been considered Systems of Systems [5] and Complex Adaptive Systems [6]. Nowadays, SDIs are being developed in many countries and are accepted as an essential infrastructure in modern societies [7, p. xiii].

Another important characteristic that has been considered for SDIs, is that they may be components of other SDIs. Rajabifard et al. [8] propose a hierarchy of SDIs, from the corporate to the global level, and point out some relationships among these levels. Masser [9] suggests that this hierarchical composition is one of the research challenges provided by SDIs. The already approved Infrastructure for Spatial Information in the European Community (INSPIRE) directive aims to build a European SDI based on the SDIs of the Member States [10].

Some aspects of the software architecture of SDIs have already been analyzed: ANZLIC [11] describes a technical architecture, services, service providers and data storage facilities, for the Internet Framework of the Australian SDI Distribution Network. Bernard et al. [12] present an architectural view of the European SDI geoportal and associated services. GeoConnections [13] describes the Canadian Geospatial Data Infrastructure Architecture following the ISO RM-ODP information, engineering and computational viewpoints. Béjar et al. [14] have proposed an architectural style, roughly correspondent to the ISO RM-ODP engineering viewpoint, for the software

components of an SDI.

Although SDIs include many different components, software architecture techniques have been mainly used to model their technical aspects. Nevertheless, some software architecture methods allow to address the non-technical components of systems too: RM-ODP provides the concepts and tools to address non-technical components of complex distributed systems, like SDIs, under the so-called enterprise viewpoint. The RM-ODP is being considered for the United Nations SDI technical governance framework, although this project is still in the design phase, and there are not many details yet [15]. It is also being used for the architecture implementation pilot of the Group on Earth Observations System of Systems (GEOSS), which objectives are related to those of an SDI [16].

Hjelmager et al. [17] have proposed an initial model for SDIs under the RM-ODP enterprise viewpoint (and also under the information viewpoint). Besides other differences, our paper improves their enterprise model in several aspects:

- We take into consideration the relationships among different SDIs and among the organizations participating in them.
- We relate policies with the interactions affected by them, consider explicitly enterprise object types and artefact role types and describe processes in UML.
- We use the recently approved ISO/IEC International Standard that establishes the use of UML to express the RM-ODP concepts [18].

In this paper, an approach to model some of the technical and non-technical components of an SDI using an architectural viewpoint is proposed. This approach allows to model SDIs as federations of autonomous organizations, where technical and non-technical components interact, under the guidelines and constraints of several policies, to achieve certain objectives. The RM-ODP enterprise language provides a set of well-defined concepts used to create an enterprise view on a system. This viewpoint addresses its purpose, expected behaviour and policies. There is also a standardized way to express these concepts as diagrams in the Unified Modeling Language (UML), and several proposals to formalize them if needed [19, 20].

The rest of the paper is structured as follows: section 2 presents a brief introduction to the ISO RM-ODP and its enterprise language. This is followed by the main part of the paper, section 3, where the elements of an

architecture to model SDIs following the enterprise language of the ISO RM-ODP are described. Finally, in section 4, some conclusions and further work are described.

2. The Enterprise Language of the RM-ODP

The ISO Open Distributed Processing Reference Model (RM-ODP) provides an architectural framework to model complex environments where heterogeneous information resources are distributed among different interconnected organizational domains [21, 22, 23, 24].

The RM-ODP allows to specify an Open Distributed Processing (ODP) system in terms of different, but interrelated, viewpoint specifications. A viewpoint on a system is an abstraction of that system addressing a particular set of concerns. Viewpoints simplify reasoning about a system, allowing its designers to focus on different concerns as needed. For the different viewpoints on a system, a viewpoint language is provided.

The RM-ODP provides five viewpoints: the *enterprise viewpoint*, concerned with the purpose, scope and policies of a system, the *information viewpoint*, concerned with the information handled by the system, the *computational viewpoint*, concerned with the decomposition of the system in objects and interfaces, the *engineering viewpoint*, concerned with the infrastructure required to support distribution, and the *technology viewpoint*, concerned with the chosen technologies used to support distribution. A complete specification of a given system would consist of several, related and mutually consistent, viewpoints. This paper is focused on the enterprise viewpoint.

The Enterprise Language of the RM-ODP defines the concepts and rules used to specify the enterprise viewpoint on a system [25]. The fundamental structuring concept for an enterprise viewpoint is that of *community*. A community is a configuration of *enterprise objects* describing a set of entities such as human beings, information resources or information processing systems, which is formed to meet an *objective*. An enterprise view must include at least one community, but it can be structured in terms of several interacting communities.

The scope (of a system) is ‘the behaviour that a system is expected to exhibit’ [25, p. 4], and it ‘is defined in terms of its intended behaviour; in the enterprise language this is expressed in terms of *roles* or *processes* or both, *policies*, and the relationships of these’ [25, p. 6, emphasis added]. These

concepts are defined later in this paper, when they are used (sections 3.3, 3.5 and 3.6).

Roles, processes, and policies allow to model the behaviour of an ODP system. The enterprise objects of a community will typically fulfil different roles at different times: the same person, an *enterprise object*, can be a user and a data producer, both of them *roles*, though not simultaneously. When this person is fulfilling the role user, she can be involved in downloading certain data set, that would be a *process*, but only if she is allowed to, for instance by a certain *policy*.

To end this section, a formal issue must be highlighted: the RM-ODP standard does not recommend any notation to specify ODP systems. Nevertheless, there is an ISO/IEC International Standard to establish how to use the UML for this task [18] (UML4ODP). In this paper that International Standard has been followed.

3. SDIs in the Enterprise Language of the RM-ODP

As described in section 2, the RM-ODP provides the necessary concepts and rules to specify distributed information systems under five different viewpoints. In the next subsections we develop an approach to facilitate the modelling of SDIs from the enterprise viewpoint of the RM-ODP.

3.1. Communities

In the RM-ODP enterprise viewpoint, systems are first specified as communities and then refined as needed. As highlighted in [8, 9], SDIs are usually composed of other SDIs, with some kind of hierarchical organization. Nevertheless, other community types are also involved: for example, two environment departments of neighbour states may agree to form a new SDI, but they are not SDIs themselves. We will say that any community that is part of an SDI is a *member* of that SDI.

We may be more precise if we take into consideration a common community type in the RM-ODP: a *federation* is a type of community formed by other communities that cooperate to achieve a common objective [25]. These communities, the federation members, are bound by the contract of the federation but they keep their autonomy. As SDIs are formed by several communities to achieve a common objective, we may model an SDI as a type of RM-ODP federation.

The relationships between SDIs and their members will typically be implemented by making some objects in the members to fulfill roles defined by the SDIs to which they belong. For instance, an SDI may require its members to have a contact point, so this SDI specifies the role ‘contact point’ and each member designates an object, e.g. a person or organization, to fulfill it. Nevertheless it may be useful to have a UML4ODP diagram which shows the relationships among the communities in an SDI without that level of detail. Figure 1 shows a very simple example with our proposal¹. That diagram includes three different SDIs (INSPIRE, Spain SDI and France SDI) and two communities which are not SDIs (Spain cadaster and the Ebro basin Authority). All communities are modelled as «EV_CommunityObjects», UML classes, which are used in RM-ODP to model communities as a whole. All SDIs must define a role which extends the corresponding ‘SDI members’ (see section 3.3.1). To show that a community is member of an SDI, it must fulfill that SDI member role. For instance, as shown in the figure, the ‘Spain cadaster’ community object is shown to fulfill the role ‘Spain SDI member’ by means of a «EV_FulfillsRole» UML association. To show which communities are SDIs, simple notes are used.

3.2. Objectives

An RM-ODP community is built to meet an *objective*. This objective may be decomposed into *sub-objectives* if needed. Communities are specified in *contracts*. An example of a community contract which includes its objective, in UML4ODP, is shown in figure 2. In that figure, the Spain SDI is modelled as an «EV_Community», a UML component, linked to its objective, a UML class, by a «EV_ObjectiveOf» UML association. The Spain SDI «EV_CommunityObject» is also shown to illustrate the use of the «EV_RefinesAsCommunity» UML dependency with the component which expresses this community.

Based on the several definitions for the term ‘Spatial Data Infrastructure’ analyzed in [2, 3], we propose that the objective of an SDI is to facilitate and promote the use of electronic spatial information resources, on a stable and supporting environment, in a geographical region where different autonomous relevant organizations² coexist, and where it is desirable, or necessary, to

¹UML4ODP does not address federations, so there is neither a recommended approach nor any hints on how to model them in UML.

²We are using organization in a broad sense, not necessarily to refer to formal or legal

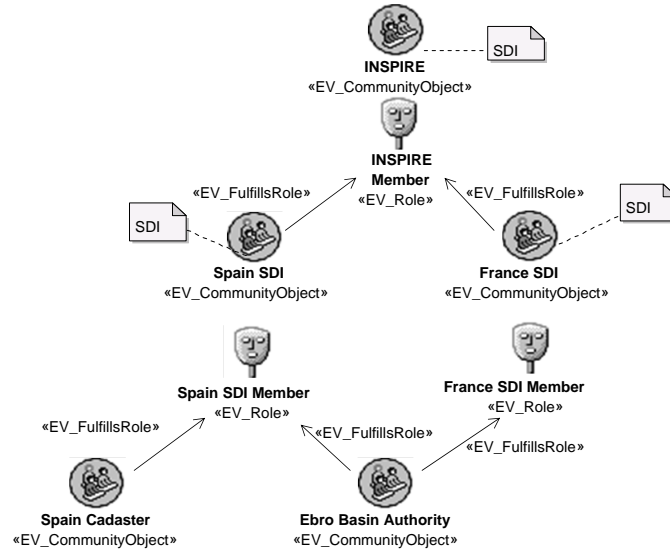


Figure 1: A diagram with an SDI and some of its members in UML4ODP

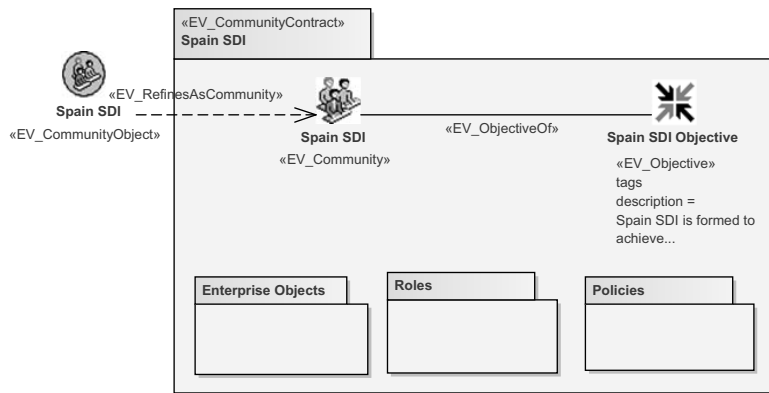


Figure 2: Spain SDI community contract in UML4ODP

keep some of that autonomy. This objective is decomposed in these three sub-objectives:

- Facilitating the creation, discovery, evaluation, exploitation, reuse, integration, and commerce of electronic spatial data and services.
- Creating a sustainable, reliable and supporting environment, by securing the necessary funds, establishing and adopting norms and policies and providing certain fundamental assets.
- Facilitating the cooperation and coordination among relevant, autonomous organizations, with different responsibilities in different areas, scales and domains.

These sub-objectives are generic and they must be considered as a starting point for concrete SDIs to specify their own.

3.3. Roles

The behaviour of a community is specified to meet its objective. It consists of the *actions* where this community objects participate. These objects participate fulfilling the *roles* defined for the community. For instance, a person can fulfill the role of user in a certain interaction, and the role of contributor in another one. Roles in RM-ODP are identifiers for behaviours (i.e. a role is a named collection of actions, with some constraints on those actions). A given object can participate in an action, as an *actor role*, or be mentioned in an action, fulfilling an *artifact role*, or can be essential for an action, requiring allocation and possibly becoming unavailable, as a *resource role*.

Roles facilitate modelling complex and scalable environments. For instance, the system administrators in two different communities in an SDI may have very different profiles and responsibilities, but when they fulfill the SDI role of *operational body*, their behaviour is well-known. This way the interactions and processes in the SDI can be defined without the need to know which objects will be participating, as long as these objects participate fulfilling the well-known roles specified for that SDI.

organizations. We use the term ‘relevant’ to refer to organizations with an interest in spatial data and services, either as producers, value-added providers or users.

The rest of this section defines some actor and artefact roles that we have found adequate to meet the objectives of an SDI. We have not found it necessary to define resource roles for the abstraction level addressed in this work.

3.3.1. Actor Roles

These are actor role types found necessary to achieve the objective of an SDI as stated before:

- **User:** They exploit the spatial assets provided by the SDI.
- **Contributor:** They contribute and/or withdraw assets, i.e datasets or services, to the SDI. A contribution is a way to make some assets available to the users (not necessarily free of charge).
- **Custodian:** They create and maintain core assets, e.g. the official topographic maps of a nation, and are responsible for its quality and availability [26].
- **Governing body:** They are in charge of creating, removing and changing policies. They also participate in the decision making activities in an SDI. This role includes characteristics of the ‘coordination body’ defined in the GSDI cookbook and [4], the ‘coordinator’ in [8] or the ‘executive level personnel’ in [4].
- **Operational body:** They are responsible for carrying out many activities in an SDI: systems administration, technical support, quality assurance or relationships among the members. This role includes, for instance, the responsibilities of the ‘catalogue administrator’ and ‘gateway manager’ in the GSDI cookbook, or the ‘operational level personnel’ in [4].
- **Contact:** They represent a community in their interactions with other SDIs. This role includes some of the responsibilities of the ‘broker’ in [17] and participates in the formal and informal engagements among SDIs described in [4, p. 188].
- **Educator:** They are responsible for the teaching and learning activities intended to cultivate the skills, technical competence, knowledge and best practices needed to maintain and use an SDI. Educators would

