CONTRIBUTIONS TO THE MODELLING OF SPATIAL DATA INFRASTRUCTURES AND THEIR PORTRAYAL SERVICES

Rubén Béjar

PhD DISSERTATION

RESEARCH ADVISORS

Pedro R. Muro-Medrano

Javier Nogueras-Iso

January 2009

Computer Science and Systems Engineering Department

University of Zaragoza

Yield and overcome;
Bend and be straight;
Empty and be full;
Wear out and be new;
Have little and gain;
Have much and be confused.

Therefore the wise embrace the one
And set an example to all.
Not putting on a display,
They shine forth.
Not justifying themselves,
They are distinguished.
Not boasting,
They receive recognition.
Not bragging,
They never falter.
They do not quarrel,
So no one quarrels with them.

Therefore the ancients say,
'Yield and overcome.'
Is that an empty saying?
Be really whole,
And all things will come to you.

Tao Te Ching (chapter 22), Lao Tzu
Source: *The Complete Tao Te Ching*
Translated by Gia-fu Feng and Jane English, Vintage Books, 1989

Acknowledgements

There are some people to whom I am grateful for their help to shape this thesis during these years. As I do not want to forget anyone, if you are not explicitly mentioned here, and you think you deserve it, then please consider yourself included too.

I would like to mention the members of the Advanced Information Systems Laboratory (IAAA) of the Department of Computer Science and Systems Engineering of the University of Zaragoza. Their constant interruptions, desperate requests for help or advice, and their need for someone who listens to their complaints about the World have been compensated by their hard work and occasional ideas.

I also want to remember the people from the National Geographic Institute of Spain (IGN). They believed in us and have continued providing their support in spite of the fact that, beyond my charming smile and good nature, I tend to question most of their proposals, to complain about everything and to give them, whenever I can, at least as much work as they give to me.

Resumen

Las Infraestructuras de Información (II) son grandes sistemas de información en red, que carecen de una autoridad central, que han evolucionado sobre una base instalada y en los que tanto los elementos sociales como los técnicos deberían ser considerados como componentes. Pueden considerarse como Sistemas de Sistemas (SoS, por sus siglas en inglés) especializados. Los SoS son sistemas autónomos y heterogéneos, geográficamente distribuidos pero interconectados, que muestran comportamientos emergentes que no eran posibles antes de que se estableciera el SoS.

Las Infraestructuras de Datos Espaciales (IDE) son II centradas en compartir datos y servicios espaciales. Los datos espaciales, o geográficos, son aquellos que describen información relacionada con alguna localización sobre la superficie de la Tierra, o con aquellas zonas adyacentes a la misma. Los servicios espaciales son entidades software, de grano grueso, descubribles y auto-contenidas, que residen en un entorno de tecnología de la información abierto, y especialmente diseñados para acceder a, y procesar, datos espaciales siguiendo una arquitectura orientada a servicios. Los servicios de representación gráfica son aquellos servicios espaciales a cargo de representar gráficamente datos espaciales, normalmente en forma de mapas, y están incluidos entre los básicos necesarios en cualquier IDE.

Esta tesis propone modelos para facilitar la especificación, el diseño y la documentación de las IDE y de sus servicios de representación gráfica. Se propone un modelo siguiendo el modelo de referencia para el procesamiento distribuido y abierto de la organización internacional para la estandarización (ISO RM-ODP, por sus siglas en inglés), para abordar varios aspectos del punto de vista de II de las IDE. Después se desarrolla un estilo arquitectural siguiendo la metodología '*Views and Beyond*' para proporcionar un patrón para ayudar en el diseño de sistemas de información que van a ser integrados en una IDE. Finalmente se presentan varias contribuciones al diseño de los servicios de representación gráfica de las IDE, y ejemplos de sus aplicaciones.

Abstract

Information Infrastructures (IIs) are large, networked information systems, without a central authority, which have evolved over an installed base, and where both social and technical elements should be considered as components. They can be considered as specialized Systems of Systems (SoS), which are geographically distributed but interconnected heterogeneous autonomous systems that show emerging behaviours that were not possible before the SoS was established.

Spatial Data Infrastructures (SDIs) are IIs focused on sharing spatial data and services. Spatial, or geographical, data describe information tied to some locations on the Earth's surface, and those zones adjacent to the surface. Spatial services are coarse-grained, discoverable and self-contained software entities, within an open information technology environment, which are specially designed to access and process spatial data following a Service Oriented Architecture (SOA). Portrayal services are those spatial services in charge of rendering spatial data, usually in the form of maps, and are included among the basic ones needed in any SDI.

This PhD thesis proposes models to facilitate the specification, design and documentation of SDIs and SDI portrayal services. A model following the International Organization for the Standardization (ISO) Reference Model for Open Distributed Processing (RM-ODP) Enterprise Language is proposed to addresses several aspects of the II viewpoint of SDIs. Then, an architectural style following the 'Views and Beyond' methodology is developed in order to provide a pattern to help the design of information systems intended to be integrated in SDIs. Finally, several contributions to the design of SDI portrayal services, and examples of their applications, are presented.

Table of Contents

Introduction

During the last decade, the growth of the Internet has made it possible to set up large distributed information systems which are not owned, nor controlled, by a single company, organization or government. When these systems present certain characteristics they are called Information Infrastructures (IIs): they are composed by social and technical components, interconnected in networks, distributed across large areas, shared by a community, enablers, i.e. they support certain tasks and applications, and usually evolve over previous systems. These characteristics make it possible to consider IIs as a type of System of Systems (SoS). An SoS is a geographically distributed system composed of several independent systems. Besides this, SoS must show some emerging behaviours and an evolutionary development.

IIs are thus substantially different from traditional distributed information systems: there is not a central authority, people (users, developers, managers...) are components of the system, its size and networked nature make it necessary to consider, and foster or constrain, emergent behaviours etc. In these systems, complete control is not possible because of the existence of autonomous components, i.e. people or independent organizations, and the lack of a central authority.

Spatial Data Infrastructures (SDIs) are IIs focused on sharing spatial data and services. Spatial, or geographical, data describe information tied to some locations on the Earth's surface, and those zones adjacent to the surface [113, p. 5]. These data are the central component of any Geographic Information System (GIS), term that is fundamentally used to refer to the use of digital data to represent space and time [113, p. 23], although there are many other aspects considered in different definitions of that term [113, pp. 5-6]. Spatial services are coarse-grained, discoverable and self-contained software entities, within an open information technology environment, which are specially designed to access and process spatial data [141], following a Service Oriented Architecture (SOA) [38]. Portrayal services are those spatial services in charge of rendering spatial data, usually in the form of maps.

A commonly cited definition of SDI is the one provided by the SDI cookbook version 2.0, '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' [78, p. 8]. Among the objectives of an SDI, this reference includes promoting a reliable environment, an infrastructure, to facilitate the access to geographical information. The agreements, organisations and programs needed to coordinate SDIs at different scales are also included. The Open Geospatial Consortium (OGC) (http://www.opengeospatial.org) and the International Organization for the Standardization (ISO) TC/211 committee (http://www.isotc211.org), have been publishing specifications and standards for spatial data services during the last decade. Many of these specifications and standards have been adopted by the different SDI initiatives in the World as a means to facilitate some of these objectives.

According to data collected during 2001 and 2002 at least 80 National Spatial Data Infrastructure (NSDI) initiatives were implemented, or in development, in the World [50]. Among the most important initiatives it is necessary to mention the one led by the Global Geospatial Data Infrastructure Association [161], the United Nations Spatial Data Infrastructure [165], the Canadian Geospatial Data Infrastructure [71], the Australian Spatial Data Infrastructure [11], the United States of America National Spatial Data Infrastructure [60] and the European directive to establish an Infrastructure for Spatial Information in the European Community (INSPIRE) [57], that, although recently approved, has already encouraged many European countries to start developing their NSDIs along the last years [56].

Currently, the Spain NSDI (Infraestructura de Datos Espaciales de España, IDEE) [89], launched in 2002 [128], coordinates 12 'regional' initiatives corresponding to autonomous communities and provinces, and including the Catalonia SDI (Infraestructura de Datos Espaciales de Cataluña, IDEC) [42] that was the first SDI initiative in Spain. Additionally at least 6 other autonomous communities are developing their own SDI, and many other institutions, i.e. municipalities, are also working on them [67].

Developing information systems is an activity where models can be used to analyze and design relevant aspects of the system in the different phases of its construction. These models, created in artificial graphical or textual languages, allow to abstract and express information of the system.

The literature provides many proposals to model SDIs, sometimes focusing on certain aspects only. The fact that SDIs are IIs, and thus should consider both social and technical components, has proven itself as a problem to develop comprehensive modelling approaches. Many of the existing models, usually focused on the technical components, are inspired by the same principles and include similar components, but this is not always easy to determine as they do not use the same terms neither the same methodologies. In addition, most of them lack solid conceptual foundations, for instance in a certain software architecture methodology, what makes them difficult to compare and use.

This PhD thesis proposes models to facilitate the specification, design and documentation of SDIs and SDI portrayal services. These models capture, refine and systematize proposals, standards, best practices and ideas found in the SDI literature and during the execution of several research and development (R&D) projects related to SDIs. The models capture knowledge about SDIs and their portrayal services that can be applied to other similar projects. More specifically, this thesis proposes models to architect SDIs as federations of autonomous communities, under some of the concerns addressed by IIs or SoS and as distributed geographic information systems that follow certain conventions and standards, and to design SDI portrayal services that follow certain tried and proven design practices.

Besides this introduction and the final conclusions, this thesis is organized in four chapters. The content of these chapters is as follows:

- Chapter 1 contains the context of this thesis. After presenting the definitions of model, pattern, software architecture and other related terms that are used in this work, SDIs are presented. A special emphasis is put in their consideration as IIs, and SoS, and their technical roots in digital libraries and service oriented architectures.

- Chapter 2 provides a model for SDIs following the ISO RM-ODP Enterprise Language, including their main social and technical components. This model gives a federated view of SDIs, presenting them as networks of autonomous communities.

- Chapter 3 presents an architectural style, under the Component & Connector viewpoint of the 'Views and Beyond' methodology, of an SDI. This style provides a conceptual template to help the designers of an information system that is going to be integrated in an SDI.

- Finally, chapter 4 presents several contributions to the design of SDI portrayal services, and examples of their application.

Chapter 1

Context and Research Issues

This chapter presents the context of this thesis. Information Infrastructures and Systems of Systems are described first, because these are the research fields that provide conceptual support for the aspects related to the main structure and organization of Spatial Data Infrastructures. Then the technical roots of SDIs are reviewed in order to narrow the problem domain of this thesis. After that, the software engineering techniques that are used along this work are reviewed, with an emphasis in the software architecture methodologies employed. Finally, the research issues addressed in this thesis are explained and their interest justified.

1.1 Information Infrastructures and Systems of Systems

As pointed out in the introduction, SDIs are usually considered as IIs focused on sharing spatial data and services. IIs provide thus a context to understand SDIs, which are the main focus of this work. In this section the definitions and characteristics given in the literature for IIs are reviewed. Then the relationship between IIs and SoS is presented in order to provide a broader context and a different approach to address the complexity of SDIs.

1.1.1 Information Infrastructures

Anne Branscomb was among the first researchers to introduce the term *information infrastructure* to refer to the media, carriers and physical infrastructure used to deliver information [35]. This term started to be common after the US plan for National Information Infrastructures was launched [166]. The European Union would launch its own plan some years later [55].

The term II would end up encompassing a broader context, including the idea of infrastructure as an enabling agent [46]. This enabling infrastructure idea is also considered in [156], where the authors point out that something is an infrastructure only in relation to some practices. They add that an infrastructure occurs when local practices are developed naturally thanks to a larger-scale technology that becomes transparent to these practices (i.e. that transparently enables them), and lists several 'dimensions' of infrastructures: they are embedded, transparent, posses reach or scope, are learned as part of membership, are linked with conventions of practice, embody standards, are built on an installed base and become visible upon breakdown.

In [81], the authors focus on information technology based infrastructures, but at the application level, not at the telecommunication networks one. They sustain that IIs are information systems, but not only information systems: they are shared by a large user community across large geographical areas and must be understood under a holistic perspective, because they are more than the individual components. They give the basic characteristics of IIs: enabling function, shared by a community, of socio-technical nature, interconnected in networks, open, heterogeneous and as an evolution of an installed base (chap. 3).

Borgman reviews several definitions for the term information infrastructure, and concludes that these can be grouped under three different perspectives: as public policy constructs that regulate communication networks and network services, as technical frameworks that incorporate the Internet and its services and as encompassing frameworks that include networks, computers, software, information resources, developers and producers [32].

II studies is a research area that addresses the technical, social and organizational aspects of infrastructures from local to global environments [33]. These authors consider that infrastructure studies should be considered as 'process building' involving simultaneously 'community-building' and 'systems-building'. II research should thus focus on integrating methods and research coming from those fields which have contributed to the study of information infrastructures: computer science, information science, communication, organization theory, cognitive science and science and technology studies.

1.1.2 Systems of Systems

Almost in parallel with the birth of the term II, it was getting clear that a new term was needed to refer to an emergent class of complex systems composed of other systems: *Systems of Systems* (SoS) [119]. Although there is not an agreed definition for the term SoS, the term has been appearing in the literature for many years. Several authors have proposed some of the characteristics that a system should have to be considered an SoS.

To distinguish large and complex monolithic systems from ´true' SoS, Maier proposes five main characteristics, sometimes referred to as Maier's criteria [119]:

- Operational independence of the elements: the component systems must be able to operate independently.
- Managerial independence of the elements: the component systems are able to operate independently, and they do operate independently.
- Evolutionary development: the SoS grows and evolves with time and experience.
- Emergent behavior: the SoS is able to perform functions that can not be found in any component system, and these functions are the main SoS objectives.
- Geographic distribution: the SoS is distributed over a large geographic extent.

Management of SoS is a challenge because of the independence of their component systems. Maier also proposes three categories for SoS, according to their strategies for managerial control, as a first step to tackle with this important issue:

- Directed: those built and managed to fulfill certain purposes. They are centrally managed and the component systems normal operations are subordinated to this central management, although they can operate independently when needed.

- Collaborative: those built and managed to fulfill certain purposes, but the component systems must voluntarily collaborate to fulfill them.

- Virtual: those who lack central management or centrally agreed purposes. Emergent, large scale, behaviours may appear, but they are not under the direct control of any central management structure.

In [51] there is another synthesis of the main characteristics of an SoS in three distinguishing traits: physical distribution; overall functionality depending on the linkages between the component systems, i.e. networks; and component system heterogeneity, i.e. people, organizations, computer systems etc. An SoS would be thus a combination of different systems that performs a function not performable by any of them alone and which shows those traits.

Other authors have also suggested a set of characteristics to differentiate SoS from other systems, focusing on the idea of composition [30]:

- Autonomy: the component systems must show autonomy, i.e. they must be free to achieve their main purpose, and this purpose cannot be being a part of other system.

- Belonging: the component systems, though autonomous systems themselves, must at the same time become a part of the SoS.

- Connectivity: the component systems must be allowed to create and destroy links among their interfaces dynamically, in order to enable the SoS.

- Diversity: an SoS should be very diverse in its capability as a system: it must be capabilities-based and not requirements-driven. This way it will be able to adapt itself to change, uncertainty and innovation.

- Emergence: because of the previous characteristics, SoS may show unexpected emergent properties, and this is desirable. The challenge is to maintain an environment where this may happen, while being able to quickly detect and destroy 'bad' emergent properties.

Although these, and other, definitions for the term SoS are different, they have points in common: SoS would be geographically distributed but interconnected autonomous systems, which include heterogeneous components, and show emerging behaviours that were not possible before the SoS was established.

SoS distinguishing characteristics make them difficult to build with traditional engineering practices. According to [63], monolithic systems depend on central control, global visibility, hierarchical structures and coordinated activities, but these characteristics can not be expected in SoS, where we find distributed control, cooperation, influence, cascade effects, orchestration and other emergent behaviours. New approaches are thus needed to tackle with this new kind of systems.

1.1.3 Information Infrastructures as Systems of Systems

The existence of a relationship between IIs and SoS has been only indirectly pointed out in the literature, probably because these terms are generally used in different research communities. For instance, in [85] they indicate that Infrastructure and Transportation Systems, which they consider under an SoS approach, are underpinned by an II, i.e. traffic control systems, that sustains them. It is also possible to find examples of certain domains, for instance health systems, where IIs [178] and SoS [175] are proposed as different approaches to similar problems.

Nevertheless, and given the previously considered II and SoS definitions, we can conclude that they share many characteristics. Some of these common characteristics are presented in Table 1.1.

There are also other issues in common. For instance, [79] presents a discussion about the tension between standardization and flexibility in IIs. IIs change during their lives, i.e. their components change, but the existence of standards, that these components used to follow and those components which are still in place continue to use, implies a resistance to changes. In [64] the authors point out that explicit and standardized interfaces are an assumption of traditional systems engineering practices, which will need to evolve to adapt to the flexibility and adaptability present in SoS.

Two of the Maier's criteria for SoS, operational independence and managerial independence do not seem to be considered, at least explicitly, in the definitions of II. Nevertheless, the fact that IIs are usually characterized as heterogeneous, networked and dependent on open and standardized interfaces would imply that it is possible that there is not a central authority, i.e. there is at least certain managerial independence of their different parts, and that these parts, interconnected thanks to their open and standardized interfaces, can also function independently, i.e. there is operational independence. Besides this, some significant examples of II, for instance the Internet itself [79], have operational and managerial independence of their components.

As the definitions for II and SoS are diverse, the relationship between these concepts is difficult to characterize. Nevertheless, it seems clear that there exists a relationship between them. For the purposes of this work we propose that IIs are a type of SoS with certain additional characteristics:

- The enabling, or supporting, nature of IIs, i.e. the idea of infrastructure, it is not considered for SoS in general. As discussed before, the idea of infrastructure may make sense only in relation with some practices that are developed thanks to its existence. An SoS can be an infrastructure, when used to support some of these practices showing those dimensions described in [156], but this is not fundamental to be an SoS.

- SoS encompass very different kinds of systems. For instance [85] mention service systems, infrastructure and transportation systems, environmental and energy systems, sensors and robotics, health systems etc. IIs are focused on the distribution of information and thus they could be 'Information Systems of Systems' or maybe 'Systems of Information Systems'.

- IIs grow over an existing installed base. SoS do not have this requirement, although the literature seems to imply that it will not be uncommon to find that they need to be developed after some of their component systems are already in place, i.e. over an existing installed base.

Table 1.1: Common characteristics found in II and SoS definitions

SoS Characteristic	**II Characteristic**
'Evolutionary development' [119]	'Evolution of an installed base' [81]
'Emergent behavior' [119], 'Emergence' [30]	Emergent changes in IIs [81, chap. 9]
'Geographic distribution' [119], 'Physical distribution' [51]	'Spatial reach' [156], 'Across large geographical areas' [81]
'Linkages between components' [51], 'Connectivity' [30]	'Interconnected in networks' [81]
'Component system heterogeneity' [51]	'Socio-technical nature', 'Heterogeneous' [81]

1.2 Spatial Data Infrastructures

The concept of *National Spatial Data Infrastructure* (NSDI) was first defined, for the United States, by the Mapping Science Committee of the National Research Council [120]. In April 1994 Bill Clinton signed an Executive Order (nr. 12906, April 11, 1994) for the establishment of the NSDI, forcing the cooperation among federal and local agencies in collecting, spreading and using geographic information. In 1996 the GSDI was created to promote global access to geographic information. Also in 1996, the Australian and New Zealand Information Council (ANZLIC, http://www.anzlic.org.au), defined the Australian Spatial Data Infrastructure (ASDI). In 1999, the Government of Canada sponsored a national partnership initiative, GeoConnections, to improve access to geospatial information and to accelerate the development of a Canadian Geospatial Data Infrastructure (CGDI, http://cgdi.gc.ca). In November 2001 the European Commission launched INSPIRE (INfrastructure for SPatial InfoRmation in Europe, http://inspire.jrc.it), an initiative to create a European directive to guide national and regional SDI development. The directive entered into force on May 15th, 2007 [57]. Nowadays, the GSDI web site lists several dozens of SDI initiatives, local, regional and national. Generally speaking, most of these initiatives have common views and objectives for SDI, as first defined by the USA NSDI, though of course they are adapted to the different realities (economical, political) of the geographic areas for what they have been established.

1.2.1 Technical Roots in Digital Libraries

In parallel to the birth of the NSDI concept in the USA, in 1994, the Alexandria Digital Library (ADL) Project began. This project intended to address some of the problems detected in map libraries. ADL project created the term *geolibrary*, defining it as "... a library containing georeferenced objects and with a search mechanism based on geographic location as the primary search key" [75, pp. 2-3]. In 1998 a panel under the aegis of the USA Mapping Science Committee conducted a workshop on distributed geolibraries, whose conclusions were published as a report by a Panel on Distributed Geolibraries of the National Research Council [138]. This report intended to update the concept of NSDI in the Internet era, with an emphasis on its foundation on distributed geolibraries. This report includes several findings related to NSDI, which point that DLs can include any kind of information with an association to a geographic place, besides the maps and images of the Earth covered by the NSDI, and that the NSDI underemphasized the importance of the geoinformation dissemination issues, which could, and should, be sustained by distributed geolibraries concepts and techniques.

As stated in the previous paragraph, over a typical definition of a digital library (DL), geolibraries add georeferenced resources and a search mechanism based in geographic location. But these, apparently simple, added elements are distinctive enough to give geolibraries several research matters of their own: based on their experience in the ADL project, [99] identify up to seven mayor issues that arise in geolibraries simply because of the special kind of content they hold: discovery of georeferenced resources, gazetteer integration, specialized ranking of search results, strong data typing and scalability, spatial context for user interfaces and the need for sophisticated geospatial resource access mechanisms.

Even with their own distinctive issues, geolibraries are still DLs, and SDIs share so many important elements with them both, that basing SDI development on the field of geolibraries, and thus on the field of DLs, provides a solid conceptual and technical foundation to build.

If DLs hold collections of digital resources then geolibraries must hold collections of digital spatial resources. But as explained before, SDIs were born with the objective to promote the creation, maintenance and distribution of geographic information. Taking into consideration the fact that most of this information is now in digital form, it is easy to recognize that one of the first roles for SDIs is very close to being a geolibrary.

Other important characteristic of digital libraries is that their resources are described by means of metadata. As geographic information resources have some unique properties (for example, location based in coordinates), there are different standards specifically designed for their metadata. Some of these standards are the evolution of library metadata adapted to the characteristics of geographic resources, like the Z39.50 application profile for geospatial metadata [130]. There are also other geographic metadata standards. One of the most important ones is the Content Standard for Digital Geospatial Metadata (CSDGM) developed for the USA by the Federal Geographic Data Committee [61], which is a national standard but it has been adopted in many other countries (South Africa, Canada...). More recently, in May 2003, the International Organization for Standardization (ISO) released the international standard 19115 [90], which includes a complex metadata schema to describe geographic information and services. These geographic metadata standards appeared in a very specific scope but they have been later mapped to other, more general, metadata standards, e. g. the Dublin Core Metadata Initiative (DCMI) proposal, via crosswalks or other technological solutions [134], in order to support interoperability.

Metadata are also of the uppermost importance in most SDI initiatives. All these initiatives show a strong emphasis in the necessity to create metadata for all pieces of geographic information, in order to leverage them to their maximum potential. [135] give a thorough view of geographic metadata in SDIs, including collections, metadata standards interoperability and heterogeneity of geographic metadata content. Applications for the creation of metadata are important to facilitate the management of any digital library. Special tools exist for geographic metadata standards like CatMDEdit [181]. CatMDEdit is an open source Java application that can be downloaded from http://catmdedit.sourceforge.net/. It supports the creation of metadata records that follow the ISO 19915 or the FGDC CSDGM metadata standards. It also supports the creation of DCMI records, emphasizing thus the relationship between different metadata standards that appeared in different scopes.

Digital libraries offer search services for the resources they hold. These services are typically based on the existence of metadata for those resources. As pointed in the previous section, a search mechanism based on geographic location as the primary search key is required in geolibraries. This is also true for SDIs, which should include catalog services that allow for this spatial search. Some of the catalog specifications for geographic data are evolutions of the search and retrieval protocols created for digital libraries. For instance, the Z39.50 protocol is included as one possible implementation for the Open Geospatial Consortium (OGC) catalog interface specification [129, 133]. This catalog interface was designed with the geospatial community in mind, but it is a generic system, that supports metadata for generic resources. It includes for example Dublin Core Metadata support, and makes it the reference for the core queryable and returnable properties. It is currently referenced as the standard catalog interface for all relevant SDI initiatives.

Another important issue with SDI catalogs is that they are designed to facilitate distributed searches. Even with the existence of a common standard for catalog interface and another one for the structure of geographic information metadata, there are still several interoperability challenges to solve. One of these challenges is the selection of appropriate vocabularies for key metadata terms, which is a problem in general digital libraries too [84]. A common approach to harmonize metadata in digital libraries is the use of controlled vocabularies, like controlled lists or thesauri. These controlled terms are typically used by catalogs to improve searches. Some existing thesauri that are useful to describe thematic geographic information are the General Multilingual Environmental Thesaurus (GEMET, http://www.eionet.eu.int/GEMET), developed by the European Environment Information and Observation Network, or the Agriculture vocabulary (AGROVOC, http://www.fao.org/agrovoc/), by the Food and Agriculture Organization of the United Nations. To facilitate the management of thesauri in the geographic context the CatMDEdit application, described before, has been integrated with a thesaurus management tool called ThManager [106].

As highlighted by [99], searching in geolibraries, requires the use of gazetteers. A gazetteer is a geographic dictionary of place names associated to their location on Earth. A natural approach to build gazetteers would be the use of thesauri, because in many cases place names, have a hierarchical organization, e. g. names of administrative units, that is naturally represented in a thesaurus. The SDIGER project, a pilot to create an SDI for water management between France and Spain, successfully followed this strategy. In this project a thesaurus for Spain and France place names was built. The place names were taken from the Spanish National Statistics Institute (Instituto Nacional de Estadística), Spanish Public Administration, and the French National Institute for Statistics and Economic Studies (Institut National de la Statistique et des Études Économiques) and organized into a place names thesaurus [189].

In order to integrate properly ontologies and thesauri in SDIs, and following their distributed, Web-services-based architecture, recent research proposes specialized Web services to manage lexical ontologies and thesauri. These services also would provide ontology-based support to other SDI components. The Web Ontology Service is a component designed to support the management and use of ontologies in SDIs [105].

The final goal of a digital library is to provide access to the managed digital resources it contains. Likewise, SDIs must provide visualization and access to geographic information, usually through specialized standard geographic Web services, like the Web Map Service (WMS), Web Feature Service (WFS) and Web Coverage Service (WCS) defined by the Open Geospatial Consortium (www.opengeospatial.org), and the ISO TC/211 committee (www.isotc211.org). These services provide access to geographic resources of different types, with different levels of granularity and with more or less processing. The WMS provides the capability to take geographic information and produce graphic maps. This capability can be used to provide a preview of geographic information resources when searching for them in an SDI catalog. WFS and WCS are Web services designed to give access to geographic information with little or no processing. An important issue with these services is that they can extract parts from geographic information resources, allowing thus a fine-grained access to them.

1.2.2 Web Services and Service Oriented Architectures for SDIs

Although the original views of SDIs include, directly or indirectly, the necessity to provide at least search, visualization and data download services [78], the trends in the evolution of distributed interoperable GIS [47, 22], based in standard geographic Web services, have made some authors argue that these services, and their architecture, chaining and orchestration, have become the fundamental component of SDIs [14]. This trend would eventually give SDIs full distributed GIS capabilities (SDI services are already being included in complex GIS analysis workflows [6]) a goal that involves not only the setup of an infrastructure to search for, and access to, geographic Web services and data, but also new design and architectural patterns, semantically aware interoperation mechanisms, and the applications that would allow final users to exploit them.

During the last years, the Open Geospatial Consortium has been working to define specifications for interoperable GIS that are recommended in most, if not all, SDI proposals. OGC work includes a discussion paper [174] describing an architecture that is a *Service Oriented Architecture* (SOA), close to the ISO 19119 [92] geographic services standard. An SOA is defined by the OASIS consortium as 'a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains' [38, p. 8]. These distributed capabilities are encapsulated as *services*, which are coarse-grained, discoverable and self-contained software entities which interact through some message-based communication model thanks to their well-known interfaces [37]. When these services are implemented following the recommendations issued by the World Wide Web Consortium (W3C), in order to make them accessible through the World Wide Web, they are usually referred to as *Web Services* [176]. OGC service specifications started before the W3C recommendations existed, and thus are not compatible with them. Nevertheless OGC is working to adapt their specifications to the W3C recommendations [155] and work is being done to facilitate this transition [153].

OGC provides Web service specifications for the basic components of an SDI pointed out in [78]: for search services, the Catalog Service Implementation Specification [129], for visualization, or portrayal, the Web Map Service Implementation Specification [100], and for data download the Web Feature Service Implementation Specification [171] and the Web Coverage Service [59]. Besides those, the OGC Website (http://www.opengeospatial.org/) lists more than 30 related specifications.

1.2.3 SDIs as Information Infrastructures and Systems of Systems

SDIs share characteristics with other kind of Information Infrastructures. The basic characteristics of IIs, described by [81], can be recognized also in the more specialized SDIs: enabling function, shared by a community, of socio-technical nature, open, heterogeneous and as an evolution of an installed base. Under this perspective, the social and technical components of SDIs are not separable and to understand them completely, it is necessary to explore the socio-technical processes and practices that can lead to a cultivated approach, in opposition to a construction approach, to SDIs [72]. This point is also sustained by other authors, who believe that SDIs are not built, but cultivated from a social and technical installed base, and enabled by IIs and information and communication technologies [131]. These authors proposals for the relationship between SDIs and IIs would be nearby to the idea of 'encompassing framework' IIs in [32]. These perspective show a different, less technological approach to SDIs, and they need to be taken into consideration for a comprehensive understanding of these infrastructures.

SDIs possess also some other non-technical, infrastructural, elements that can be considered characteristic for them. First of all, SDI efforts are mainly sustained by public administrations. There are laws and regulations (the INSPIRE directive in

Europe [57] or the USA NSDI executive order (nr. 12906, April 11, 1994)), public funding for SDI [9], issues related to the access to public information and e-government [180], and the recommendation of political sustainability and a legal framework for the success of NSDIs [9]. In this subject, they are nearer to the traditional public infrastructures (i.e. transport networks) and related to the 'public policy' IIs in [32].

In section 1.1.3 it has been proposed that IIs are a type of SoS with certain characteristics. Given that SDIs are a type of IIs, we may conclude that SDIs are a type of SoS, as pointed out, although not developed, by the United Nations SDI proposed technical governance framework [13]. If SDIs are a type of SoS, the techniques and solutions that we can apply to create, maintain and study SoS can be used, maybe after some adaptation, for SDIs too. Systems of Systems Engineering (SoSE), and Architecting (SoSA), can provide several of these techniques and solutions.

A very significant example of SoSE process is presented in [41]. In that work, a complete, architecture-centric and model based SoSE process is developed. This process addresses the necessity to go from high level 'mission objectives', those that justify the existence of the SoS, to requirements, constraints and system-level functionality of the different components of that SoS. The process is architecture-centric in the sense that the SoS architecture is the main artefact used to conceptualize, construct, manage and evolve it. Because of the special characteristics of SoS, the architect must focus on interoperability and system interfaces to balance performance and risk. The integration of general system capabilities and particular customer goals is managed using a structured method that, through the SoSE process, leads to the central architecture model. The process is exemplified with the Global Earth Observation System of Systems (GEOSS) [110], that as a net-centric, all-volunteer organization-driven SoS with contributing data systems dropping in and out of it all the time, requires a flexible and robust development approach that could be applicable to SDI development too.

Management and governance are related issues. They present important challenges in SoS because of the managerial independence of their component systems and the lack of a central authority. Starting from the SoS characteristics proposed in [30], listed in section 1.1.2, in [76] there is a proposal for an SoS Operational Management

Matrix that helps to identify management best practices that could be applied to SoS management in general, and thus to SDI management in particular. In [127], starting from the study of the governance issues in service-oriented architectures, they propose six characteristics for good SoS governance:

- Collaboration and authority: since an SoS has not a single owner, governance will be collaboratively created by the participating organizations, which will follow it because it is in their own interest.
- Motivation and accountability: as it is difficult to enforce policies in a context without a single central authority, public performance measures are proposed as a way to prevent poor behaviour of the involved systems.
- Multiple models: given the variety of SoS, it is foreseen that different governance models will be developed for different SoS.
- Expectation of evolution: as evolution will be usual in SoS, some actions will be needed to tackle with it: informing about changes, coordinating schedules when changes affect others, maintaining several versions of the component systems to keep compatibility with older systems, taking measures to isolate systems from changes and minimizing the perturbations to interfaces when a system is changed.
- Highly fluid processes: planning for rapid changes in SoS governance is needed, although taking into consideration localities in the SoS: it is easier to agree to a change of the governance rules affecting a small 'neighbourhood' in the SoS, than changing governance rules that affect the entire SoS.
- Minimal centrality: although SoS will be generally distributed, there may appear centrality when there is a dominant system in the SoS, or when certain infrastructure is required for the SoS to function. Anyway this should be kept to a minimum. These characteristics could be immediately applicable to SDI governance, although their implementation remains as a research challenge.

The effectiveness of an SoS in fulfilling its objectives is difficult to assess. In [64] they suggest several areas that influence the effectiveness of network-centric SoS, those enabled by and built upon large-scale communication networks, like SDIs, and propose several conditions about their desirable future state:

- Social and cultural environment: the environment should motivate collaborative behaviour.
- Legal and regulatory framework: there should exist a legal framework that recognizes and regulates SoS.
- Management practices: they should be sufficiently defined and performed for SoS.
- Governance procedures: they should be cooperative and distributed.
- Engineering practices: they should be appropriate for developing and evolving SoS.
- Technology base: it should support the realization of the network-centric vision.

These areas can be taken as a base to develop processes and techniques to set up SDIs, and the conditions about their desired state could be refined to develop metrics to asses the state of those SDIs already in place. On the other hand, there are several methodologies to evaluate SDIs, in [77] they review some of them, or to estimate their cost [132], which could provide a base to generalize them to the evaluation of generic, network-centric SoS, what remains as an unsolved problem as shown here.

One of the characteristics listed in section 1.1.2 for SoS is that they should be capabilities-based and not requirements-driven. Nevertheless they are usually built with a purpose and some objectives, and these need to be analyzed and modelled to verify they are fulfilled. In [111] the authors present an ontology-based framework that allows them to integrate different requirements engineering techniques (goal-driven, scenario-based, viewpoint-based etc.) in software intensive SoS. With this framework, objectives and goals for a system can be expressed in different ways, according to different stakeholders at different levels of detail, and then integrated. Through the use of the so-called Multi-Dimensional Link Analysis on the ontology described before, these authors suggest that some emergent behaviours of the system can be predicted and controlled. Although this solution only addresses the system at the requirements level, i.e. when it is being established, it tackles with one of the most complex problems that SoS and SDIs have to address: the existence of unavoidable, some times undesirable, emergent properties.

1.3 Models and Patterns

The Institute of Electrical and Electronics Engineers (IEEE) Computer Society Guide to the Software Engineering Body of Knowledge (SWEBOK) [1], uses the term *model* for several different artifacts produced during software development. As this PhD thesis deals with software models, it is important to clarify the use of that term along this work.

In [1, p. 2-6], *conceptual models* are mentioned as a way to understand real-world problems by representing entities from the problem domain and their real-world relationships and dependencies. Object or data models, among others, are suggested. Notations, and the processes which guide their application, are used to represent conceptual models. UML (Unified Modeling Language) is given as an example of widely accepted notation [151]. After this, the SWEBOK sustains that *architectural design*, the description of the structure, organization and components of a software, is closely related to conceptual modeling: they are different topics that broadly use similar notations and methods.

With regards to software architecture, the SWEBOK (p. 3-3), in the software design chapter, defines it as a description of the subsystems and components of a software system and their relationships. It also recognizes that software architecture has emerged as a broader discipline during the last decade, and that it has brought several concepts, *architectural style*, *design pattern* etc. that can be seen as ways to capture and describe generic design knowledge. Different software design notations, both graphical and textual, are used to represent the different software design artifacts.

This PhD thesis is focused on the kind of models pointed out in the previous paragraphs for the domain of SDIs and SDI portrayal services: descriptions in a graphical notation, possibly complemented with textual information, of the structure, organization and components of SDIs and SDI portrayal services. These descriptions capture and express design knowledge related to those domains.

The models in this thesis are convenient and accepted ways to present reusable design knowledge related to certain domains, and are not related to any specific software engineering methodology or process. Thus, it is expected that they could be used to guide the creation of either a Computation Independent Model (CIM) in a Model Driven Architecture ®(MDA) specification [126], or just some agile models [7]. How the proposed models could be employed in these practical situations is beyond the scope of this work.

Regarding the term *pattern*, several authors provide definitions. According to [40], they present generic solutions to recurring problems in specific contexts. In [66], a pattern is considered an idea that has been useful in a practical context and could be applied to other situations.

The SWEBOK distinguishes between macroarchitectural patterns, i.e. architectural styles, and microarchitectural patterns, i.e. design patterns. The difference is their scope: macroarchitectural patterns describe the high-level organization of a software while microarchitectural patterns describe details at a lower level. In both cases, they are pointed out as a strategy to guide the software design process (p. 3-5).

Another possible distinction is between design and analysis pattern. A *design pattern* is a recurring solution to a common problem in software design [69]. An *analysis pattern* is a group of concepts that represent a common construction in business modelling (i.e. at the conceptual level) [66].

Patterns can be seen as 'template models': descriptions of the structure, organization and components of a given solution to a generic problem, with enough information to apply them properly (i.e. in which context they have been proven useful, the rationale behind them, known uses, etc.). These 'template models' represent analysis and design knowledge that can be tailored to apply it to specific systems, with specific instances of the generic problems solved by the patterns. They are abstractions above the level of single classes, instances or components, which provide a common vocabulary for analysis and design principles [40].

1.4 Software Architecture

According to [103], the term *software architecture* involves the structure and organization by which components and systems interact to form systems, and also the properties of these systems that are best designed and analyzed at the system level. In [45], the authors sustain that it is the structure of structures of a system, formed by elements, their visible properties and the relationships among them. The IEEE [88] indicates that the architecture of a system is its fundamental organization, embodied in its components, their relationships, among them and to the environment, and the principles which guide its design and evolution. Although there is not a complete agreement, these definitions, and others, point out that a software architecture is the structure, or organization, of a system and is formed by elements, some times called components, their properties and the relationships among them and possibly with their environment.

Nevertheless, there is another use of the term software architecture. The IEEE Guide to the Software Engineering Body of Knowledge [1] cites the definition proposed by [40, p. 384]: 'software architecture is a description of the subsystems and components of a software system and the relationships between them'. The International Organization for Standardization (ISO) and International Telecommunication Union Telecommunication Standardization Sector (ITU-T) Reference Model for Open Distributed Processing (RM-ODP) defines the architecture of a system as 'a set of rules to define the structure of a system and the interrelationships between its parts' [94]. In these definitions, software architecture is an artefact of the software design process.

The term software architecture for a system is thus defined either as an inherent property of this system, or as an artefact that specifies or describes certain properties of the system. Along this work the term architecture is sometimes used with the meaning of 'design artefact' in chapter 2, because that is the meaning provided by the methodology used in that chapter, ISO RM-ODP, but in general the 'inherent property of a system' meaning is implied. Anyway, an effort has been done to use the terms description, model, documentation, specification etc. along with the term architecture when the meaning 'design artefact' is intended and the context alone is not enough to understand which meaning is used.

Architecture is design, but only the design concerned with the visible properties of those components which are part of the fundamental structure of a software system. Architects draw the line deciding which decisions must be taken so that the system fulfills its main purposes, and which decisions may be taken by the designers of the different components. If the design of a component involves deciding its structure and sub-components, then that design is also architectural, but only from the point of view of the designer of that component [45].

Expressing or documenting the architecture of a system may be a very complex task. The approach that has been consolidated over the last years is expressing the software architecture of a system as a set of *views*, each of them addressing different concerns for different users. Indeed, documenting an architecture without specifying the type of view that is being used tends to create too complex diagrams, with too much information and without a clear separation of concerns.

The 'Views and Beyond' proposal in [45] and the IEEE recommended practice for architectural description of software-intensive systems [88], share a similar approach that allows to describe the architecture of a system as a set of views which follow some defined *viewtypes*, *viewpoints* in the IEEE standard, and styles. The viewtypes define what can, or must, be included in a view and what not. Other proposals even prescribe a fixed set of views that should be given for any system, such as the 'Siemens four views' described in [87] or a fixed set of viewpoints such as the ISO and ITU-T Reference Model for Open Distributed Processing (RM-ODP), presented in [144]. Architectural *styles* are refinements of the defined viewtypes, and capture some commonly occurring forms and variations. Architectural views, but also viewtypes and styles, can be created in different ways: with diagrams, more or less structured text descriptions, or even formally with different Architecture Description Languages (ADLs) [125].

1.4.1 Views and Beyond

The 'Views and Beyond' software architecture methodology described in [45] proposes to document the architecture of a software system in different views. These views represent some of the elements of the system, and their relationships. As different views support different objectives, they do not recommend a fixed set of views. Instead of that, they provide a set of viewtypes, that are definitions of the element and relationship types that can be used in a certain view. They also provide a set of styles, which are specialization of the element and relationship types in the viewtypes, and constraints on their use. The architects of a system choose which views they need. This methodology also provides some guidelines to extend the set of provided viewtypes and styles.

The proposed viewtypes are three: the module viewtype, to partition the system in code units with certain responsibilities, the component-and-connector viewtype, to define views with elements that have a runtime presence (i.e. processes, objects, data stores etc.) and the allocation viewtype, that maps software onto its environment (i.e. hardware, network, file systems etc.). For each of these viewtypes several styles are defined in the methodology.

1.4.2 ISO Reference Model for Open Distributed Processing

ISO and the IEC (the International Electrotechnical Commission), in collaboration with the ITU-T, prepared the International Standard ISO/IEC 10746 with the general title 'Information technology — Open Distributed Processing — Reference Model'. This standard is written in four parts [96, 94, 93, 95], and explained in detail by [144].

The ISO Open Distributed Processing Reference Model (RM-ODP) provides an architectural framework to specify distributed information systems. It allows to model complex environments where heterogenous information resources are distributed among different interconnected organizational domains.

The RM-ODP allows specifying an ODP system in terms of different, but inter-related, 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. Viewpoints are not phases in the design of a system: they are simultaneous views on that system. For each viewpoint, a viewpoint language based on a set of common concepts 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 specification of a given system would consist of several viewpoints following the different viewpoint languages. Besides this, there must be explicit relationships and correspondences among these viewpoints, and they must be mutually consistent.

RM-ODP does not recommend any notation to specify ODP systems. Nevertheless, there is an ISO/IEC Final Draft International Standard to define the use of the Unified Modelling Language (UML) for this [98]. In this work the recommendations in that draft have been followed in the diagrams.

In order to facilitate reading the diagrams in this work, here is a brief summary of the rules proposed by that ISO draft to map UML elements to RM-ODP enterprise language concepts: there is a UML profile for each RM-ODP viewpoint. In these profiles, stereotypes are used to express the RM-ODP concepts as UML elements. For instance, enterprise object types, a concept in the enterprise viewpoint of the RM-ODP, are represented as UML classes stereotyped as «EV_Object»; role types, another RM-ODP concept, are also represented as UML classes, but stereotyped as «EV_Role»; since enterprise objects may fulfill roles, this is expressed in UML with associations between them stereotyped as «EV_FulfillsRole» etc. UML allows to use icons to represent stereotypes and the ISO draft proposes a set of icons: we have chosen to use both the icons and the text stereotypes to facilitate readability. Generalization/specialization relationships, multiplicities etc. are represented as usual in UML.

1.5 Research Issues

As shown in the introduction of this thesis, SDIs are implemented, or in development, in many countries around the World, and there are several initiatives, coming from public administrations and from other organizations, to set up international SDIs too. In some countries, e.g. in Spain, there are also several regional and local SDIs already in place. Nevertheless, there are yet many unresolved problems related to these infrastructures, as pointed out by several authors [15, 74, 123].

The field of SDIs has its roots in geographic information systems, digital libraries, service oriented architectures and information infrastructures. The field of systems of systems provides another viewpoint for SDIs, because it is related with information infrastructures as shown in section 1.1.3. Contributions to the field can come from any of these areas and from other related ones.

This thesis addresses problems related with the modelling of SDIs and their visualization components under different perspectives. These problems have been studied in the literature, however significant research contributions are possible:

- Under the perspective provided by information infrastructures, and systems of systems, an SDI is composed of socio-technical components that interact in an environment without a centralized authority. Although the II point of view has been addressed to some extent in the literature [46, 72, 131], some authors have developed categorizations for the different social and technical components of an SDI and their interactions [147, 172], and the hierarchical composition of SDIs has been studied [148, 123], there are not many attempts to provide a systems point of view that can address all these issues for SDIs. ISO RM-ODP has been proposed, and used to a certain extent, to model SDIs [86, 13], but these proposals are yet immature and incomplete. Chapter 2 is devoted to this problem.

- Under a software components and services point of view, there are many proposals that describe the elements of an SDI. Some authors have pointed out the services that an SDI must provide [78, 14] while others have proposed architectures in a service oriented style [91, 58, 70, 17]. These models usually lack a foundation on

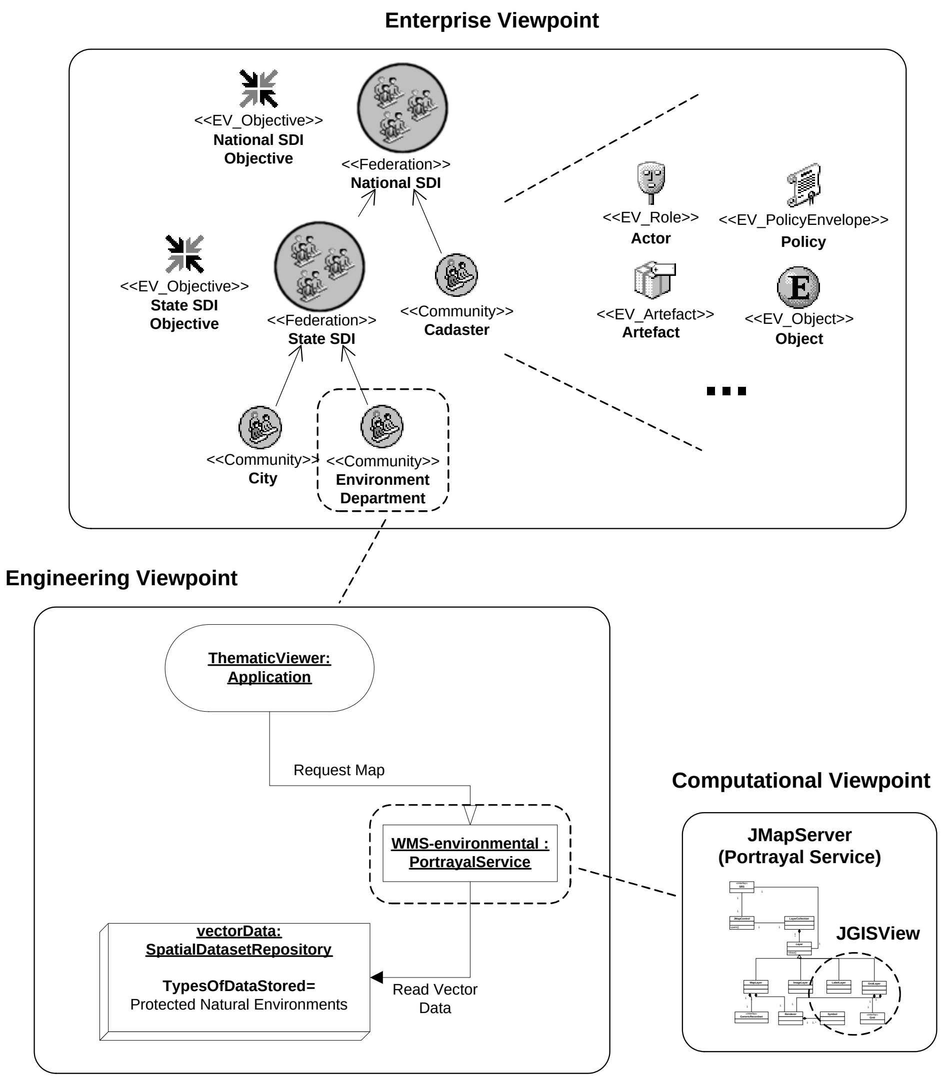

Figure 1.1: Different viewpoints on an SDI

a well-known software architecture methodology. Another problem is that they share many components, but differences in notation and terminology makes it difficult to compare them. Chapter 3 proposes an architectural style to deal with these issues.

- For the main technical components, i.e. Web services, of SDIs, there is a high degree of consensus in using OGC and ISO specifications for their interfaces. Nevertheless, how these interfaces are implemented is left to their designers. Portrayal services, those used to display spatial data, are among the basic services proposed for SDIs. Different strategies for the implementation, design and architecture of these services have been studied in the literature [102, 191, 101, 154]. It has been also researched how these services can be used to support the development of applications which include GIS capabilities [16, 122, 2, 116, 118]. Chapter 4 describes several models for SDI portrayal services.

Figure 1.1 relates the main chapters in this thesis using the Enterprise, Engineering and Computational viewpoints in the ISO RM-ODP methodology (see section 1.4.2). This does not mean that this thesis intends to provide complete and RM-ODP compliant views of any SDI. The RM-ODP viewpoints are used here just because they are a convenient way to arrange the different problems addressed in this thesis, which can be briefly described as different techniques to facilitate the modelling of SDIs at different levels of detail: chapter 2 would address SDIs as federations of communities (Enterprise Viewpoint), chapter 3 would be concerned with the components in one of those communities (Engineering Viewpoint) and chapter 4 would help to address several analysis and design issues in one of those component types (Computational Viewpoint).

Chapter 2

A Model for Spatial Data Infrastructures in the Enterprise Language of the RM-ODP

2.1 Introduction

There are many definitions, architectures and models for Spatial Data Infrastructures (SDIs), and they share many similar elements. The inclusion of people, institutional frameworks and standards among these elements shows the necessity to have a perspective on SDIs that can integrate their social and technical aspects. Some authors have pointed at Information Infrastructures (IIs) as this perspective.

The present chapter proposes an architectural pattern to model SDIs under some of the concerns addressed by IIs. This pattern allows to model SDIs as open systems composed of different organizations, people, and heterogenous and networked data and services.

The proposed pattern expresses SDI concepts in the enterprise language of the International Organization for Standardization Open Distributed Processing Reference Model (ISO RM-ODP). As the RM-ODP includes five viewpoint languages, including the enterprise language, the pattern can be seen as a first step towards an ISO RM-ODP Domain Specific Reference Architecture for SDIs.

In the second part of this chapter, this architectural pattern is applied to the case of the Infrastructure for Spatial Information in the European Community (INSPIRE).

This allows to verify the applicability of this architecture to a complex and real SDI.

The rest of the chapter has five parts. Firstly, we examine previous works that have proposed definitions and models for SDIs. The next section presents a brief introduction to the ISO RM-ODP. This is followed by the main part of the chapter, where an architectural pattern to facilitate the modelling of SDIs following the enterprise language of the ISO RM-ODP is detailed. After that, the pattern is tested by applying it to the case of INSPIRE. The final section draws some conclusions and proposes further research.

2.2 Previous Work

There are many definitions for the term SDI, some of the most relevant ones are cited in [147], but in general they share common objectives and similar components. Indeed, several authors, after reviewing different definitions for the term SDI, have proposed similar categories for its main components: pursuing an early, working definition for the term Global Geospatial Data Infrastructure, [46] suggest people, policies and agreements, standards and technologies. [147] concluded that the main components of SDIs could be grouped in five categories: data, access networks, policy, standards and people. [172, p. 22-23], takes these categories and proposes a framework composed of five core components: data, people, institutional frameworks, technology and standards, where the institutional framework component includes aspects related with people and policies, and the technology component covers aspects related to access, policies, standards and data. [180] suggest that the political aspects are more critical than the technical ones for the consolidation of SDIs. These and other works show the necessity to address very different concerns, from people to technology, to provide a full picture of SDIs.

Another important characteristic that the bibliography has considered for SDIs is that they are distributed and may be components of other SDIs. [148] propose a hierarchy of SDIs, from the corporate to the global level, and point out some relationships among these levels. [123] suggests that this hierarchical composition is one of the research challenges provided by SDIs. Other authors have considered SDIs as special cases of IIs [46, 15, 74], while others have indicated that at least these are related concepts [131]. In [80] several properties of IIs are highlighted: their social and technological elements cannot be separated, they are shared resources for a community, integrated through standardized interfaces, open and heterogenous. This relationship with IIs thus reinforces the ideas of distribution and networked composition for SDIs.

Some aspects of the software architecture of SDIs have also been analyzed in the bibliography: [10] describes a technical architecture, services, service providers and data storage facilities, for the Internet Framework of the Australian SDI Distribution Network. [17] present an architectural view of the European SDI geoportal and associated services. [70] describes the Canadian Geospatial Data Infrastructure Architecture, following the ISO RM-ODP information, engineering and computational viewpoints. [26] have proposed an architectural style, roughly correspondent to the ISO RM-ODP engineering viewpoint, for the software components of an SDI.

The previous paragraphs show that 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 IIs and SDIs, under the so-called enterprise viewpoint. An initial model for SDIs under that viewpoint has been proposed [86]. Although this model is just briefly described, does not define most of the elements included, does not address the relationships among different SDIs, and does not clarify the use of UML to model the RM-ODP concepts, it is an step in that direction. RM-ODP is also being considered for the United Nations SDI technical governance framework; however this project is still in the design phase and there are not many details yet [13].

2.3 SDIs in the Enterprise Language of the RM-ODP

An ODP system is a distributed information system, possibly deployed in an environment of heterogenous information technologies and different organizational domains, and conform to the RM-ODP requirements.

In this chapter, we are modelling SDIs as ODP systems. This does not imply that

SDIs can not be considered under other paradigms. It just provides us with a solid foundation to develop a detailed model of relevant aspects of an SDI.

As described in section 1.4.2, the RM-ODP provides the necessary concepts and rules to specify ODP systems under different viewpoints. In the next sections we develop an approach to use the enterprise language of the RM-ODP to model the enterprise viewpoint of SDIs. As this viewpoint is concerned with the purpose, scope (expected behaviour), and policies of a system, only these concerns are addressed.

This chapter does not present the specification of a given SDI; it presents a set of concepts that intend to contribute to the task of defining a domain specific reference architecture for SDIs [144, p. 173], from which software architectures for concrete SDIs could be derived. These concepts, defined after reviewing relevant literature, can be seen as a pattern to specify the purpose, scope and policies of different SDIs.

In order to achieve this, all the elements presented in the next subsections represent types, and not instances. Following the object oriented approach used by the RM-ODP, these types can be refined and instanced as needed to specify concrete SDIs. Section 2.4 provides an example of application of this pattern to help to clarify its use.

2.3.1 The Enterprise Language of the RM-ODP

The Enterprise Language of the RM-ODP defines the concepts and rules to specify a system from the Enterprise Viewpoint. This viewpoint is concerned with the purpose, scope, and policies of a system. The scope is defined as the expected behaviour.

Although the Enterprise Language is described in [93], there is a more recent document which refines and extends it [97]. In this work, the latest version is used.

An enterprise specification of a system describes that system and parts of its environment, always from the perspective of its purpose, scope and the policies that apply. The fundamental structuring concept for an enterprise specification is that of *community*. A community is a configuration of *enterprise objects* describing a set of entities, human beings, information resources, information processing systems etc. that is formed to meet an *objective*. An enterprise specification includes at least one community, but it can be structured in terms of several interacting communities.

The scope of the system is expressed in terms of *roles*, *policies*, *processes* and their relationships. The definitions of these concepts are presented later in this work, when they are used (sections 2.3.4, 2.3.6 and 2.3.7).

2.3.2 Communities

In the RM-ODP enterprise viewpoint, systems are first specified as communities and then refined as needed. The first step to model an SDI is thus deciding if other types of communities are needed, besides the SDI itself.

As described in section 2.2, SDIs are usually defined as possibly composed of other SDIs, with some kind of hierarchical organization. But there are many scenarios where other community types are involved: for example, two environment departments of neighbour states may agree to form a new SDI, but they are not SDIs themselves; or a university with a large archive of historical maps may decide to join its national SDI, to share them. We will say that any community that is part of an SDI is a *member* of that SDI, but the term *node* will be used sometimes to stress the networked nature of SDIs.

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. These communities, the federation members, are bound by the contract of the federation but they keep their autonomy. Depending on the contract, these communities may even be allowed to withdraw from the federation. As SDIs are formed by several communities to achieve a common objective, we may model an SDI as a type of RM-ODP federation.

To show the relationships among the members of a given SDI, we can represent SDIs as directed graphs of interrelated communities, as shown in the example of figure 2.1[1]. In that graph, vertices are SDIs and other communities, and the directed edges represent membership relationships from members to SDIs; if there is a directed path between two vertices, then there is a membership relationship between them: for instance, 'company A' is a member of 'city A SDI' because there is an edge between them, but it is also

[1]Note that the icon used to represent SDIs is not provided by the draft for using UML discussed in 1.4.2 because it does not include federations

a member of 'state A SDI', because 'city A SDI' is a member of 'state A SDI'. This implies that the objects belonging to any community will have to conform to the rules of all SDIs that, directly or indirectly, include that community.

To improve readability, and to highlight the hierarchical structure of SDIs, the graph in figure 2.1 is vertically arranged. Any community may be a member of an SDI, but other community types can not have members themselves. This does not imply that these communities can not be composed of other smaller communities: this is possible, but it would not be shown at this abstraction level.

This graph shows that although there is a general hierarchical organization, there are situations that require to be more flexible. For instance, 'state B environment department' is a node of 'state B SDI', but it is also a node of 'global environment SDI'. This situation could happen if the country where state B belongs does not have an environmental SDI, but state B does and wants to join the 'global environment SDI': this situation may be seen as undesirable, or not, but it is possible and should be taken into consideration.

As SDIs are increasingly being established to share not only spatial data, but also services and other assets, the nature of the possible relationships among them is changing: a given community will typically be interested in the spatial data of its neighbours, and also in the spatial data that covers its area at different scales, but it may be interested too in interacting, and exchanging services and other assets, with communities that are 'thematic' neighbours, i.e. they share interests in a problem domain, but not 'physical' neighbours. This may lead to complex patterns of interaction among SDIs and their member communities that would need to be researched.

The relationships between member communities and SDIs will typically be implemented by making some objects in the communities to fulfill roles defined by the SDIs where they belong. For instance, an SDI may require its communities to have a contact point, so this SDI specifies the role 'contact point' and each member designates an object, e.g. a person or a team, to fulfill it.

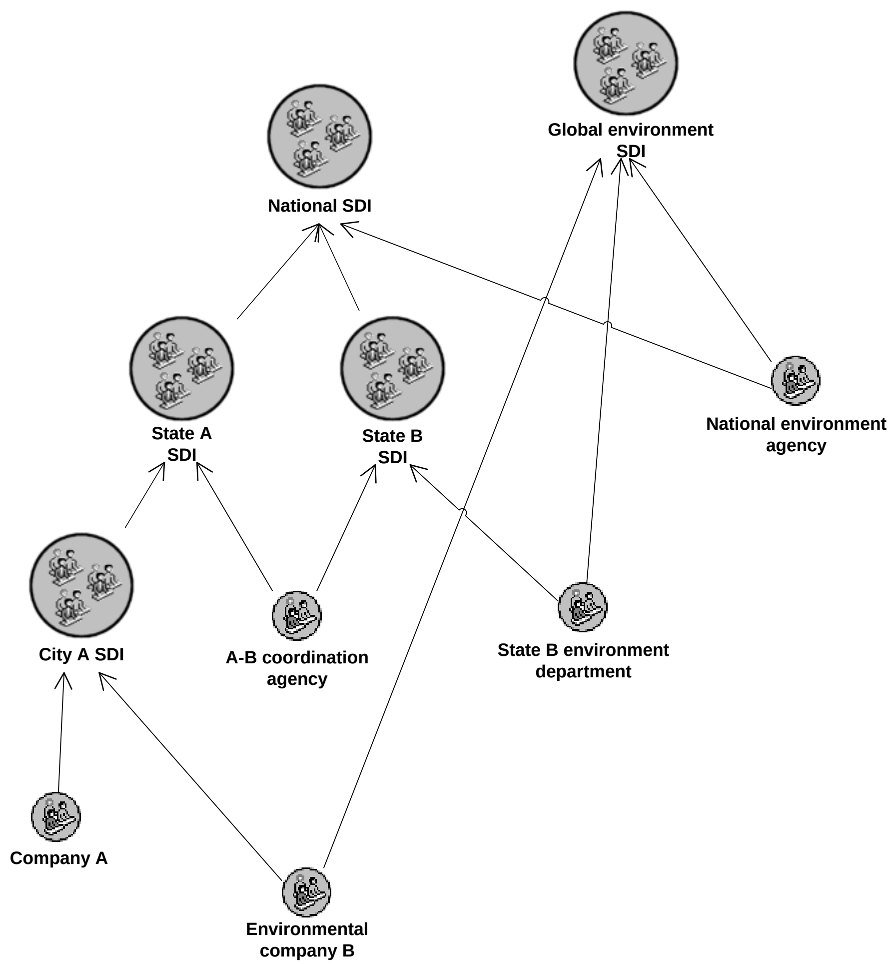

Figure 2.1: SDIs as a graph of interrelated communities

2.3.3 Objectives

An RM-ODP community is built to meet an *objective*. This objective may be decomposed into *sub-objectives* if needed. Although different SDIs have different objectives, these objectives usually have common elements. After analyzing the literature referenced in section 2.2, we will consider in this chapter that an SDI is a federation of communities built to facilitate and promote the use of spatial information resources, on a stable and supporting environment, in a certain extent where different autonomous relevant organizations coexist, and where it is desirable, or necessary, to 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 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 mentioned many times in the following pages, usually to justify the necessity of the different components presented. For the sake of readability, from now on these mnemonics will be used to refer to them: '*spatial asset availability*', '*infrastructure creation*' and '*cooperation & coordination*'.

The aforementioned sub-objectives are interrelated, but this classification has proven itself useful to express and refer to them in this work. We are considering that a community that does not address all these sub-objectives should not be considered a 'full' SDI, although a system with a subset of these sub-objectives could show some of the characteristics of SDIs, or could be considered one in progress. For instance, a community that intends to achieve 'spatial asset availability' and 'infrastructure creation' could be a centralized, corporate GIS; a community that intends to achieve 'infrastructure creation' and 'cooperation & coordination' would be an information infrastructure, without the spatial focus; a community that addresses 'spatial asset availability' and 'cooperation & coordination' should not be considered an infrastructure, i.e. long term, enabling, trustworthy..., unless it is also addressing, at least to a minimum extent, 'infrastructure creation' etc. This position is arguable, because there are existing definitions of the concept SDI which do not consider all these sub-objectives, but after analyzing the literature reviewed in section 2.2, we have decided to require those. Anyway, those objectives are very generic and thus they must be considered as a starting point for concrete SDIs to specify their own ones.

To finish this section, it is important to clarify a few points that we do not intend to imply with the proposed objectives:

- There may be some organizations that decide to set up an SDI without a legal mandate, so a legal status is not mandatory.

- We are using organization in a broad sense, not necessarily to refer to formal or legal 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 process to establish an SDI is not addressed in this work. It may come from a legal mandate, i.e. 'top down', or it may come from groups of users that decide to organize themselves, i.e. 'bottom up', or it may be a hybrid process or something completely different.

- A single organization may establish an SDI internally, but only if it is composed of sub-organizations, e.g. departments, that keep some autonomy to take decisions related to the SDI. If this is not the case, i.e. there is a single central authority, we consider that this organization has not established an SDI. It may have established a system with some characteristics of SDIs, or it may become a member of an SDI, but that would not be an SDI as defined here.

- 'Infrastructure creation' does not necessarily depend on public resources or a legal mandate. It may include a long-term vision, commitment, formal or informal plans, or mechanisms to obtain the resources needed for a supporting and

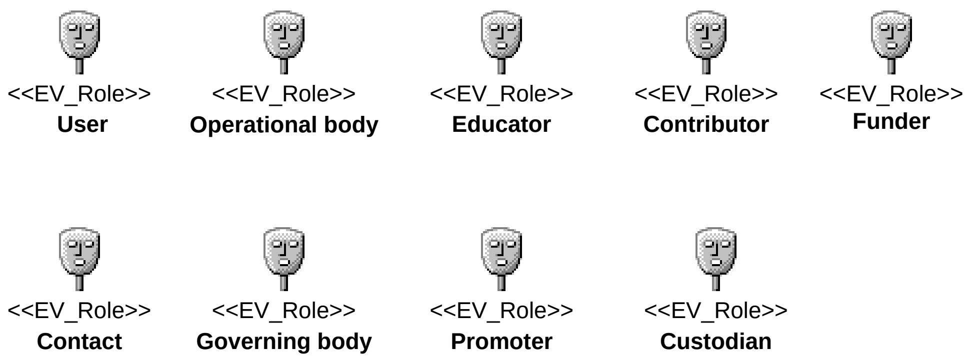

Figure 2.2: Actor Role Types in UML

trustworthy environment, but these may come from public authorities, private companies, user associations etc.

2.3.4 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. A given object can participate in carrying out an action, as an *actor role*, or be mentioned in an action, fulfilling an *artefact role*, or can be essential for an action, requiring allocation and possibly becoming unavailable, as a *resource role*. Roles allow us to have objects which behaviour is different in different circumstances. For instance, a person can act as a user in a certain interaction, and as a contributor in another one. This section defines some actor and artefact roles that we have found adequate to meet the objectives of an SDI. We have not found necessary to define resource roles for the pattern presented in this chapter. Refining this pattern to apply it to a concrete SDI could require more detailed interactions, that could require some resources to be defined, but at the level of abstraction used, the more general artefact roles have been enough.

Actor Roles

The actor roles are shown in UML in figure 2.2. They are described in table 2.1, although it is important to take into consideration the next sections (i.e. interactions where they participate, policies that affect them etc.) to clarify their place in an SDI.

Table 2.1: Actor role types in the enterprise viewpoint of an SDI

Actor role	Description	Rationale
User	They are the main beneficiaries of the SDI.	This role is needed to define many interactions and processes in an SDI, specially those related with the sub-objectives 'spatial asset availability' and 'infrastructure creation'.
Contributor	They contribute and/or withdraw the assets, i.e. datasets or services, they own or control. A contribution is understood as a way to make some assets available to the users of an SDI, i.e. they are findable and there is a way to get or use them. It does not require the assets are for free and it may be necessary to get a license from the contributor.	Contributors posses some of the characteristics of several actors mentioned in the GSDI Cookbook and other SDI references: 'contributor to the catalogue', 'data producer', 'product provider', 'service provider' etc. Any of these actors may play the role of contributor, but they could have other responsibilities too (i.e., a 'data producer' also creates and possibly sells data, but that does not depend on the existence of an SDI). They help to achieve 'spatial asset availability' and 'infrastructure creation' (when providing core assets).

Table 2.1: Actor role types in the enterprise viewpoint of an SDI

Actor role	Description	Rationale
Custodian	They create and maintain core assets, and are responsible for its quality and availability.	Described in [162], they help to achieve 'infrastructure creation'.
Governing body	They are in charge of creating, removing and changing policies. They also participate in the decision making activities in an SDI, for those activities not regulated by any policy.	This role includes characteristics of the 'coordination body' in the GSDI cookbook and [172], the 'coordinator' in [148] the 'policy maker' in [86] or the 'executive level personnel' in [172]. The governing body helps to achieve 'infrastructure creation' and 'cooperation & coordination'.
Operational body	They are responsible for carrying out most activities in an SDI: systems administration, technical support, quality assurance, relationships among the members etc. They enforce policies, and initiate, or respond to, some processes and interactions.	This role includes, for instance, the responsibilities of the 'catalogue administrator' and 'gateway manager' in the GSDI cookbook, or the 'operational level personnel' in [172]. They participate in every subobjective of the SDI.

Table 2.1: Actor role types in the enterprise viewpoint of an SDI

Actor role	Description	Rationale
Contact	They represent a community, not necessarily an SDI, in their interactions with other SDIs, and with the members of those SDIs.	This role will have some responsibility in the coordination activities mentioned in most SDI references: for instance, it would include some of the responsibilities of the 'broker' in [86] or would participate in the formal and informal engagements among SDIs described in [172, p. 188]. They are fundamental to achieve 'cooperation & co-ordination'
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.	Providing education on the SDI is considered by most SDI initiatives. Capacity building is pointed out as a characteristic of the current generation of SDIs for instance by [146]: educators would hold responsibilities on information and training for capacity building as described in [73]. It does not need to be a 'formal' educator: any user my fulfill this role when sharing his/her experiences with other users. Educators help to achieve 'infrastructure creation', by contributing to capacity building and providing a supporting environment.

Table 2.1: Actor role types in the enterprise viewpoint of an SDI

Actor role	Description	Rationale
Promoter	They are responsible for publicizing an SDI, components, objectives, benefits etc., and for keeping the different actors informed of news and changes.	The promotion of the SDI is an activity mentioned in the GSDI cookbook. Promoters help fundamentally to achieve 'infrastructure creation', by helping to provide a supporting environment, and also to achieve 'cooperation & coordination' by helping the different actors to be informed of news and changes that can affect the coordination activities.
Funder	They provide the funds needed to keep the SDI.	The GSDI cookbook highlights the importance of funding, gives some examples for different SDIs, and makes some suggestions in order to ensure funding and persuade funders (p. 110-112). It would be possible, and maybe common, that people or organizations that are funders, are given also other roles in an SDI (e.g. they participate in a governing body). Funders are needed in order to achieve 'infrastructure creation'.

The literature provides us with other potential SDI actor roles. Given the concerns addressed by the enterprise language of the RM-ODP, and the objective of our work, we considered a number of them that finally were not included. We find especially interesting, and very real, the concept of 'white knight' proposed by [49]. We have not

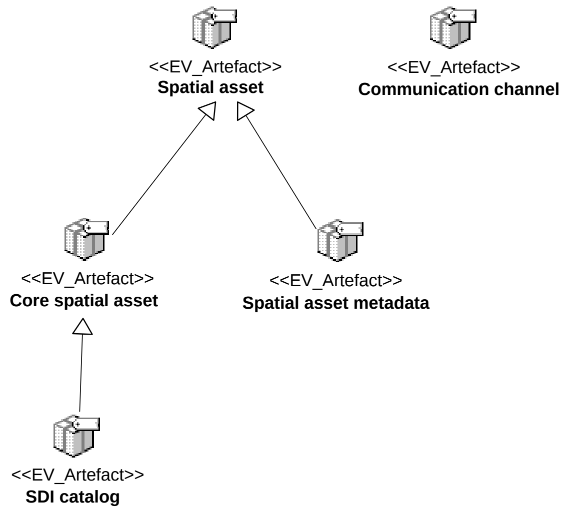

Figure 2.3: Artefact Role Types in UML

been able to model white knights as such: a white knight would some times act as promoter, as educator, as a governing body etc., but its real role is almost intangible and difficult to capture. We have also decided not to model the role of an 'SDI leader'. We have included a generic 'governing body' that concrete SDIs will refine, possibly including some kind of leader.

Artefact Roles

The artefact roles, those referenced in actions, are shown in UML in figure 2.3. They are described in table 2.2, although it is also important to take into consideration their context to clarify their place in an SDI.

Table 2.2: Artefact role types in the enterprise viewpoint of an SDI

Artefact role	Description	Rationale
Spatial asset	Any useful or valuable spatial information resource that can be made accessible to the users of an SDI.	It is a generalization used in interactions and processes where different spatial information resources (datasets, applications, services, educational or promotional stuff...) are needed only because they may be of interest to some SDI actors, and not because their main function. They are fundamental to achieve 'spatial asset availability'.
Core spatial asset	A refinement of a spatial asset, used to model those spatial assets that are part of the foundation of an SDI, i.e. those essential to achieve its objectives.	The GSDI cookbook describes 'consistent reusable themes of base cartographic content (framework, fundamental, foundation or core data)' (p. 10). The core spatial asset would be a generalization that would include also core services, support applications, data models, etc. They are fundamental to achieve 'infrastructure creation' and also to facilitate 'cooperation & coordination' by providing a common foundation for different organizations to work on.

Table 2.2: Artefact role types in the enterprise viewpoint of an SDI

Artefact role	Description	Rationale
SDI catalog	The catalog of an SDI where every spatial asset metadata can be found.	Every SDI initiative considers catalogs in some way or another: it can be a single service, or a gateway to a distributed network of services. It may be called register, directory or clearinghouse. It is fundamental to achieve 'spatial asset availability' and one of the fundamental resources that must be provided to achieve 'infrastructure creation'. If implemented as a Web service, it would correspond with the 'catalog service' in [26]
Spatial asset metadata	A type of spatial asset that provides information, i.e. a structured description, about another spatial asset.	It is a fundamental element in every SDI initiative. Its existence makes it possible to achieve 'spatial asset availability'.

Table 2.2: Artefact role types in the enterprise viewpoint of an SDI

Artefact role	Description	Rationale
Communication channel	The means used by the actors of an SDI to exchange information, and to access the spatial assets.	It is needed to achieve every SDI subobjective: 'spatial asset availability', 'infrastructure creation' and 'cooperation & coordination'. Common examples would be a geoportal, Web forums, mailing lists etc. Less common examples would be an e-marketplace or some e-commerce services used to facilitate access to some assets that require a payment. It can be seen as a consequence of the distributed nature of SDIs, which require the existence of ways to keep the different communities and actors connected. The technical details of the communication channels in an SDI are not relevant from the enterprise viewpoint.

2.3.5 Enterprise Objects

Enterprise objects model entities that are needed in the specification of a system from the enterprise viewpoint. These entities can be human beings, legal entities, software components, data resources etc. All enterprise objects fulfill at least one role in at least one community. They participate in actions fulfilling these roles. The same enterprise object may fulfill different roles at different moments.

As the behaviour of the enterprise objects depends on the role they fulfill, we have found that only a few enterprise object types are needed for the pattern presented in this chapter. In addition, these object types are quite generic, i.e. they may appear in very different RM-ODP communities.

Some enterprise object types that can fulfill the actor roles presented in section 2.3.4, are listed in table 2.3. In that table some very simple definitions are provided just to clarify their use in this work, and the differences among them. These objects must be taken as examples, as many other types could fulfill the aforementioned actor roles.

Table 2.3: Enterprise object types for actor roles in the enterprise viewpoint of an SDI

Enterprise object	**Rationale**
Person	An individual human being. They can fulfill the roles of user, contributor, governing body, operational body, funder, contact, educator and promoter.
Team	A group of people, usually small, with a common objective. They can fulfill the roles of user, contributor, governing body, operational body, funder, contact, educator and promoter.
Organization	A stable entity formed by people with a certain purpose, and guided by a set of, typically formal, rules. It can fulfill the roles of user, contributor, governing body, operational body, funder, contact, educator, promoter and custodian. This is the only object type that we have found appropriate for custodianship, as this activity would require long term commitment and, possibly, a formal institution.

A number of enterprise object types that can fulfill the artefact roles presented in section 2.3.4, are described in table 2.4. These objects are more specific than those provided for the actor roles, but again they must be taken as common examples of object types that can fulfill the artefact roles, and not as fixed list.

Table 2.4: Enterprise object types for artefact roles in the enterprise viewpoint of an SDI

Enterprise object	Rationale
Dataset	A collection of data. They can fulfill the roles of spatial asset, core asset and spatial asset metadata (if it describes another asset).
Spatial dataset	A collection of data related to geographic locations. They can fulfill the roles of spatial asset and core asset.
Spatial application	A software system that allows users to perform a set of tasks, mainly related with spatial data and metadata, possibly accessing to some spatial services. They can fulfill the roles of spatial asset and core asset. They correspond with the 'applications' in [26].
Spatial service	A software system, with an interface for other software systems, that provides operations to access to, or work with, spatial data and metadata. They can fulfill the roles of spatial asset, core asset and SDI catalog. They correspond with the 'SDI services' in [26].
Geoportal	A Web site mainly focused on spatial content, spatial services, and the tools to discover them. They can fulfill the roles of spatial asset, core asset, SDI catalog (i.e. if implemented as database accessible through the geoportal), and communication channel. They correspond with the 'Geoportal' in [26].

2.3.6 Policies

Policies are used to identify specifications of, and constraints on, behaviour that can be changed during the lifetime of a system, or that can be tailored to apply to different systems. Policies can be used to parameterize a specification establishing behaviours that are dependant on certain policies that are not fully specified. These policies need to be detailed in order to completely model a system that follows that specification. For example, the specification for a certain family of systems may require that a ‘quality policy’ is applied to certain processes, but the details of this ‘quality policy’ in specific systems may differ.

In this work, we propose a set of policies we have found relevant to model the behaviour of SDIs. These polices will differ, maybe substantially, among different SDIs, but in some way or another, any SDI following the architectural pattern proposed here will need to implement most of them. These policies are shown in UML in figure 2.4 and described in table 2.5.

Policies control the behaviour of roles, and the development of processes and interactions. Some controlling relationships are quite straightforward: the education policy controls the behaviour of educators, the promotion policy controls the behaviour of promoters, the governance policy controls the behaviour of the governing bodies etc. How policies control interactions and processes is described in section 2.3.7.

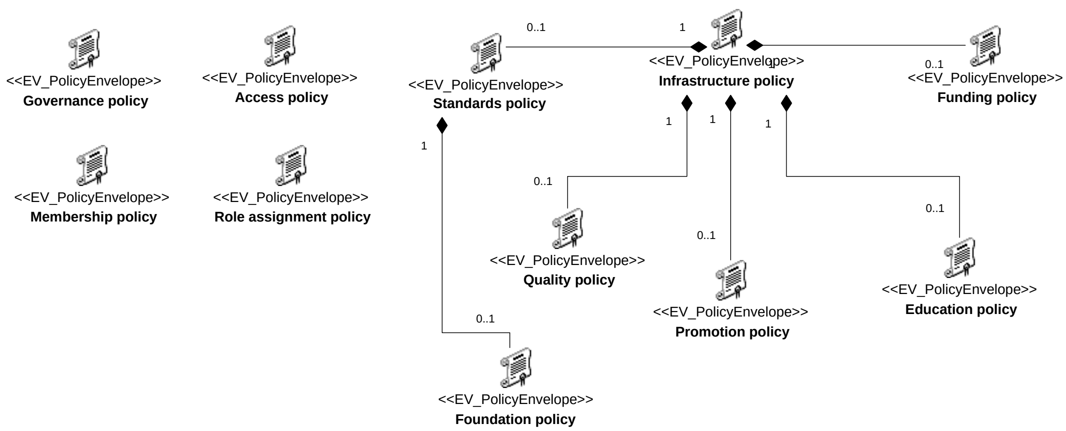

Figure 2.4: Policies in UML

Table 2.5: Policies in the enterprise viewpoint of an SDI

Policy	Description	Rationale
Governance policy	The rules that regulate the decision making and policy making activities in an SDI.	This policy, or a similar one, would be needed for any community with a minimum size or complexity. It could be refined, because decision making and policy making involves many different and important activities, but that would be very dependant on the type of SDI: its size, its scope, if its legally mandated or not etc.
Role assignment policy	The rules that establish the enterprise objects that may fulfill the different roles, and under which circumstances. This policy is needed in almost every RM-ODP community.	Some simple rules are given in tables 2.3 and 2.4, with the different types of enterprise objects described there. Refinements of this policy can vary a lot among different SDIs.
Infrastructure policy	The rules that help to enforce that an SDI and its components, processes, members etc., possess certain properties that contribute to make them a stable, reliable and supporting environment.	Needed to achieve every sub-objective of the SDI.

Table 2.5: Policies in the enterprise viewpoint of an SDI

Policy	Description	Rationale
Standards policy	The rules that facilitate the exchange of information and services, by the specification or adoption of certain norms.	Needed to achieve every sub-objective of the SDI.
Foundation policy	A part of the standards policy, this policy establishes the core assets of an SDI.	Needed to facilitate 'spatial asset availability' and to achieve 'infrastructure creation'
Quality policy	The rules that guide quality assurance processes in an SDI.	Needed to achieve 'infrastructure creation'.
Promotion policy	The rules that foster and guide the activities that make publicity for an SDI.	This policy regulates the behaviour of the promoters and helps to achieve 'infrastructure creation' and 'cooperation & coordination'.
Education policy	The rules that foster and guide the teaching and learning activities in an SDI.	This policy regulates the behaviour of the educators and helps to achieve 'infrastructure creation'.
Funding policy	The rules that establish how the necessary funding to keep an SDI is secured.	This policy may potentially influence every sub-objective of an SDI. The behaviour of funders is affected by this policy.

Table 2.5: Policies in the enterprise viewpoint of an SDI

Policy	**Description**	**Rationale**
Access policy	The rules that establish the mechanisms to access and withdraw spatial assets in an SDI.	This may include property rights management, licensing, price policies, the rights that members keep over the spatial assets they contribute (i.e. whether after a contribution they keep the right to withdraw it) etc. It regulates 'spatial asset availability' and may also affect 'infrastructure creation' (i.e. if it requires that accessing certain assets costs money, that is then used to support the SDI).
Membership policy	The rules that regulate the relationship among an SDI and its members: rights and obligations, entry and exit procedures, etc.	This policy determines 'cooperation & coordination'.

Some authors have indicated that there may exist a 'legal mandate' to establish an SDI, for instance [78, p. 100]. Besides this, most, if not all, SDIs will have to explicitly consider the legal framework that affects them. Although these legal mandates and legal frameworks could be modelled as policies, they exist before and outside the SDI, so they have not been modelled as SDI policies in this work. Nevertheless, it will be necessary to include elements from these legal mandates and frameworks into the policies of most SDIs.

2.3.7 Interactions and Processes

Enterprise objects participate in *actions* fulfilling roles. If two or more objects participate in an action, or when a single object interacts with itself, it is said to be an *interaction*. *Processes* specify how collections of actions take place to achieve some result. Interactions provide us with a way to express the collective behaviour of communities, while processes provide us with a way to focus on the achievement of certain objectives.

RM-ODP supports different approaches to specify behaviour: roles, interactions, processes, or a combination of them. For this work, we have decided to express the behaviour of SDIs using mainly roles and interactions, although some common processes are briefly described too.

The interactions we have found necessary to achieve the fundamental objectives of most SDIs are shown in figures 2.5, 2.6, 2.7 and 2.8; they are detailed in table 2.6.

We have also described some common behaviours that could be modelled as processes in table 2.7. Those processes are only some examples: most SDIs will have many more, but we consider processes may be very different among different SDIs, so they are not good candidates for expressing a pattern.

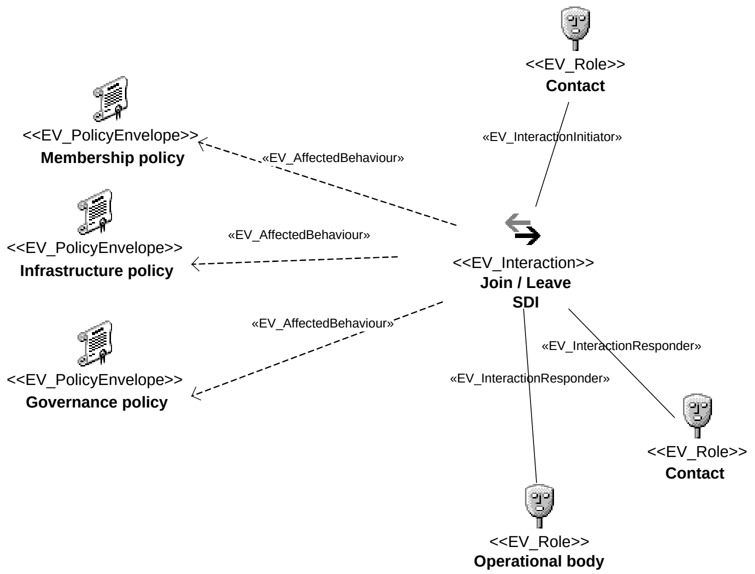

Figure 2.5: Join and Leave SDI interactions in UML (shown in one diagram, but they are two different interactions)

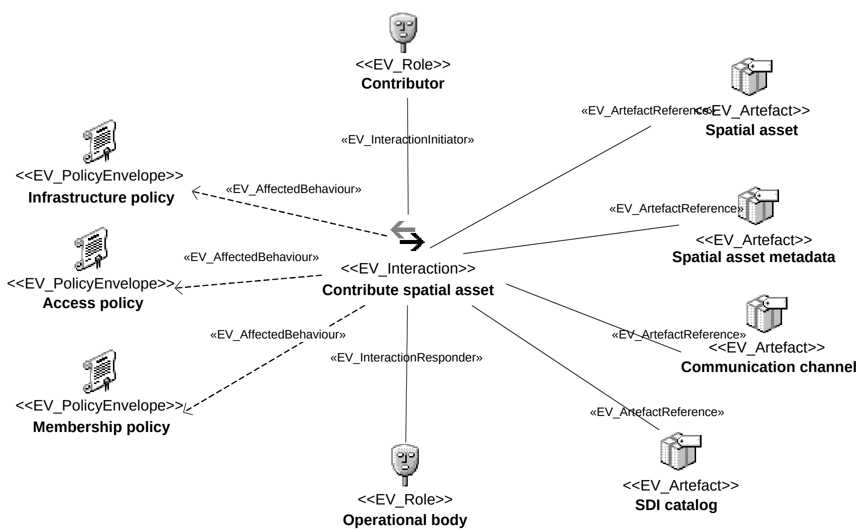

Figure 2.6: Contribute spatial asset interaction in UML

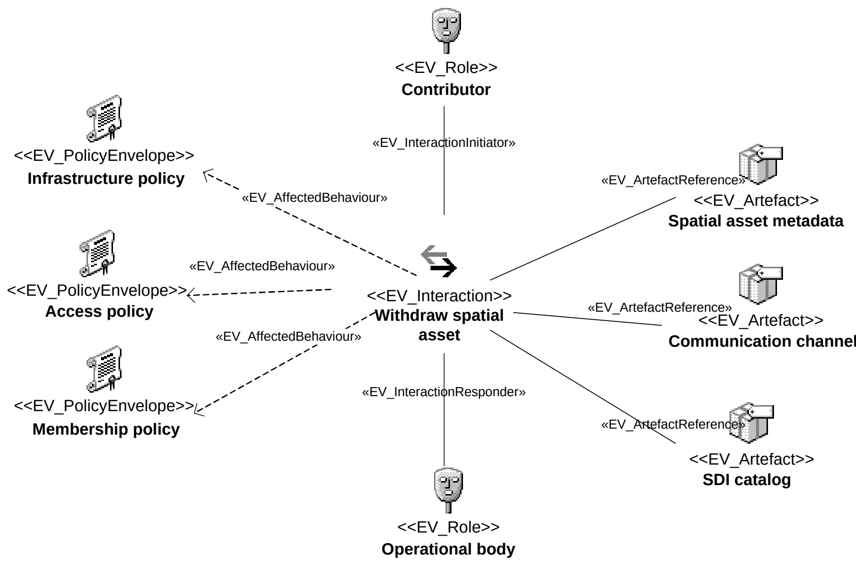

Figure 2.7: Withdraw spatial asset interaction in UML

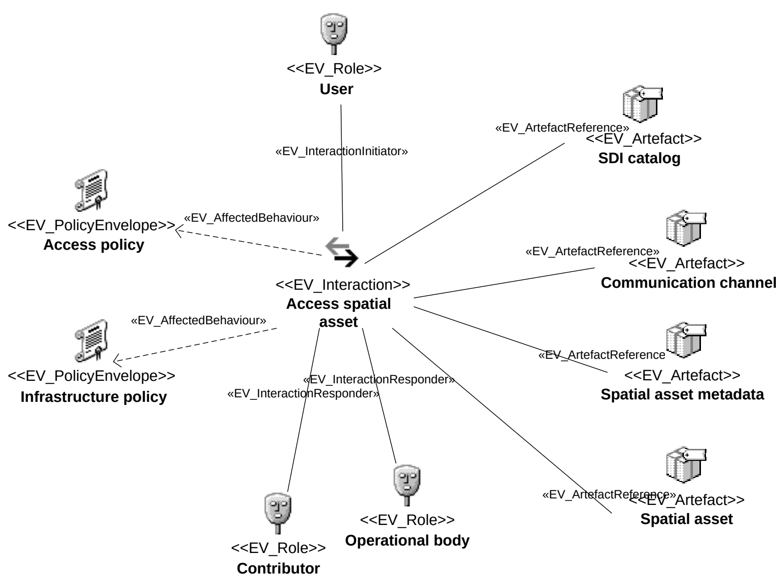

Figure 2.8: Access spatial asset interaction in UML

Table 2.6: Interactions in the enterprise viewpoint of an SDI

Interaction	**Description**	**Rationale**
Join SDI	An SDI, represented by a contact, joins another SDI, represented also by a contact. The interaction is regulated by the governance, the membership and the infrastructure policies, and approved by an operational body of the SDI that is incorporating the new member.	The membership policy establishes the requirements to the joining member to be admitted, and the governance policy the procedures to follow. The infrastructure policy may include certain indirect requirements: for instance, if certain quality parameter is expected to be achieved by all members, the new member must comply. How the 'first contact' is achieved is not modelled, i.e. there is not any communication channel artefact, because that is considered to happen outside the SDI (after all, the joining community is not a member of the SDI yet). It is essential to achieve 'cooperation & coordination'.
Leave SDI	This interaction is similar to the Join SDI interaction described above, although it occurs when a community wants to leave an SDI.	The membership policy determines the requirements that a member must fulfill to withdraw from an SDI, and if that is permitted. It is natural to consider this interaction if joining an SDI is also considered. It is essential to achieve 'cooperation & coordination'.

Table 2.6: Interactions in the enterprise viewpoint of an SDI

Interaction	Description	Rationale
Contribute spatial asset	Any contributor with a spatial asset, and the will to contribute it to the SDI, contacts an operational body through an SDI communication channel. The contributor must provide the spatial asset metadata, that the operational body will include in the SDI catalog.	This interaction is regulated by the infrastructure and membership policies, that may indicate certain requirements a spatial asset must fulfill in order to be accepted as a contribution to the SDI, and also certain requirements for the contributor (i.e. maybe different member types have different requirements for their contributions). The access policy allows the operational body to check if the contribution will be accessible as required by the SDI (i.e. free of charge, with an acceptable price or license, etc.). It is essential to achieve 'cooperation& coordination', 'spatial asset availability' and 'infrastructure creation' when the contributions are core assets.

Table 2.6: Interactions in the enterprise viewpoint of an SDI

Interaction	**Description**	**Rationale**
Withdraw spatial asset	Any contributor with certain rights over a spatial asset, and the will to withdraw it from the SDI, contacts an operational body through an SDI communication channel. The contributor must provide the spatial asset metadata, in order o identify the spatial asset to be withdrawn. The operational body will remove the metadata from the SDI catalog.	This interaction is regulated by the infrastructure and membership policies, that may indicate that certain assets from certain members can not be withdrawn, or can be only withdrawn after certain requirements are fulfilled (i.e. other member is providing an equivalent spatial asset). The access policy may indicate if the contributor has the right to withdraw the spatial asset. It is essential to achieve 'cooperation & coordination' and 'spatial asset availability'.

Table 2.6: Interactions in the enterprise viewpoint of an SDI

Interaction	Description	Rationale
Access spatial asset	Any user with an interest in a spatial asset, uses the SDI catalog to obtain the spatial asset metadata in order to find out the requirements to access it. These requirements may involve contacting an operational body, e.g. to get technical support, or the contributor responsible for that asset, e.g. to obtain a permission. This contact would happen through an SDI communication channel.	The access policy exists essentially to regulate this interaction: who can access what and how. The infrastructure policy may regulate some quality aspects in the access to certain assets, or some rules regarding the access to core assets. This interaction generalizes how the users gain access to the spatial assets in an SDI. Gaining access may be downloading a dataset, perhaps after some payment, finding out the information to use a Web service, and maybe getting a permission, etc. It is not intended to model simple operations such as making a request to a well-known Web map service: the enterprise viewpoint of the RM-ODP does not address that level of detail. This interaction is essential to achieve 'spatial asset availability'.

Table 2.7: Processes in the enterprise viewpoint of an SDI

Process	**Rationale**
Establish/change policy	It specifies how a governing body can change a policy, or create a new one. It is controlled by the governance policy. Refinements of this process would include adopting new standards, i.e. changing the standards policy, adopting changes in the laws affecting the SDI, i.e. changing the governance or the membership policies etc.
Assure quality/standard	It specifies the steps an operational body must take in order to verify that a spatial asset has the quality required by the quality policy, or complies with the norms specified by the standards policy. It is controlled by the infrastructure policy. This process would be applied, for instance, as a part of the contribute spatial asset interaction.
Best practice	This generic process includes those that have been labelled as 'best practice' in an SDI. For instance, setting up a Web map service with certain software may be described as a best practice process. This process would not be required for any member, but could be useful for some of them. There are too many activities that would fit this description to try to list them here, but we find it important to include these best practice processes, because they contribute substantially to achieve 'capacity building', and thus to the sub-objective of 'infrastructure creation'.

2.4 Application to INSPIRE

The European directive for the establishment of the Infrastructure for Spatial Information in the European Community (INSPIRE) entered into force on May 15th, 2007 [57]. This text establishes rules aimed at the establishment of a European SDI, built upon the European Community Member State SDIs.

INSPIRE provides us with a significant example of a complex SDI where very different aspects are addressed: data, technical components, complex organizational issues, interoperability and harmonization, intellectual property etc. The INSPIRE directive is a 'high level' law: on the one hand it will be refined by the so-called implementation rules (technical arrangements, specifications etc. that will supplement several elements in the directive); on the other hand it will be transposed to the different Member State laws, possibly being extended and adapted in the process. Nevertheless, in its current state it is already complex enough to be a useful test.

In this section, the feasibility to use the proposed pattern to model the main components and rules addressed by INSPIRE is tested. We are not proposing a model of a European SDI following this pattern. We are just verifying the possibility to create such a model, by testing its applicability to the concepts and rules in INSPIRE.

In the rest of this section, we will analyze INSPIRE following a similar structure to the one presented in section 2.3. References to articles in INSPIRE will be made as Art. X (meaning Article X of the directive). The concepts defined for the pattern in section 2.3 are shown **in bold** to facilitate readability.

2.4.1 Communities and Objectives

The INSPIRE directive establishes a European Community SDI built upon the SDIs operated by the European Community Member States. It also takes into consideration the fact that the Member States will establish their SDIs according to their different levels of government. Finally, it also highlights the role that some European Community institutions and agencies must play. Figure 2.9 exemplifies how to express a possible arrangement of INSPIRE following the guidelines established in section 2.3.2, although it must be noted that INSPIRE does not regulate how the Member States must set up

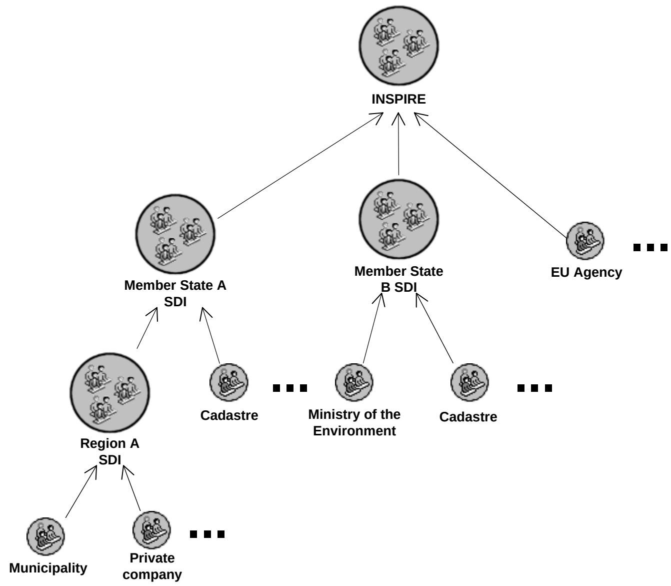

Figure 2.9: INSPIRE as a graph of interrelated communities

their SDIs, so the final picture will depend on how the Member States transpose this directive to their own laws.

The objectives of INSPIRE are compatible with those proposed in section 2.3.3:

- It is created to facilitate sharing and reuse of spatial data and services.
- It intends to be reliable and sustainable, i.e. it is legally binding for the Member States and requests continuous monitoring and periodic reporting.
- It requires an effort in coordination among the different Member States, but also among their own public authorities.

2.4.2 Actor Roles

After analyzing the text of the INSPIRE directive, certain actor roles have been identified. Some of them are similar to those proposed in section 2.3.4: **users** are mentioned several times and **custodians** are partially implied in Art. 14.2. Other actor roles

66

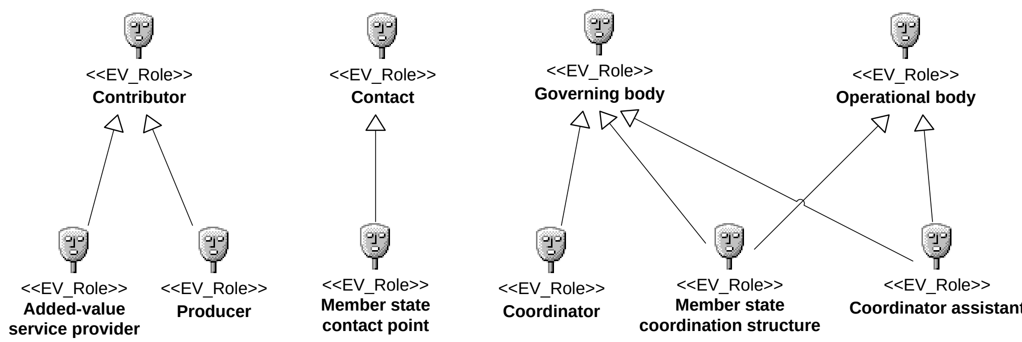

Figure 2.10: INSPIRE actor roles that refine those in figure 2.2

could be considered refinements of those in section 2.3.4: these are shown in figure 2.10.

Added-value service providers and producers, mentioned several times in the directive, are refinements of the **contributor** role. The Art. 19 describes the following roles: Member State contact point, a subtype of the **contact** role, INSPIRE coordinator, a role to be fulfilled by the European Commission, Member State coordination structure, and coordinator assistant, a role to be fulfilled at least by the European Environment Agency. Although quite undefined, we have considered that the coordinator is a subtype of **governing body** and that the Member State coordination structure and the coordinator assistant are both subtypes of **governing body** and **operational body**.

It is important to notice that the Member State coordinating structure, although required by INSPIRE, will be a **governing body** in that Member State SDI, but not necessarily in the European SDI.

2.4.3 Artefact Roles

Analyzing the INSPIRE directive, we have concluded that the set of artefact roles defined in section 2.3.4 can be used to model INSPIRE artefacts from the Enterprise

Viewpoint of the RM-ODP. This set could be extended, but it includes the basic necessary concepts:

- Although **spatial assets** do not exist explicitly in INSPIRE, spatial data sets and spatial data services, at least, would fulfill this role.

- Arts. 8 and 11 present some fundamental resources that can be modelled as **core spatial assets**.

- The INSPIRE geo-portal, defined in Art. 15, is a **communication channel**, an **SDI catalog** (supported by some discovery services (Art. 11)) and it may be considered also a **core spatial asset**.

- Metadata are fundamental in INSPIRE, there is a whole chapter of the directive for them (chapter II), and correspond to **spatial asset metadata**.

2.4.4 Enterprise Objects

As explained in section 2.3.5, the pattern includes some generic enterprise objects that can fulfill the roles defined to achieve the objectives of an SDI. Besides these, the INSPIRE directive includes several other elements that can be modelled as enterprise objects too:

- Spatial data sets, spatial data services and the INSPIRE geo-portal are very similar to, or subtypes of, those objects with similar names proposed in section 2.3.5.

- Public authority may be a subtype of **organization** and/or **person** with certain public responsibilities.

- A third party may also be a subtype of **organization** and/or **person**.

- European institutions and bodies can be modelled also as subtypes of **organizations**, or possibly as a subtype of public authority.

- The directive mentions explicitly the EU Commission and the European Environment Agency, that would be modelled as object instances of European institutions.

2.4.5 Policies

As a European Community directive, INSPIRE is full of rules that can be modelled as policies. As this directive is a legal mandate to establish a European SDI, it is expected that most of these rules will be included, and possibly extended, in the policies ruling that SDI. Table 2.8 shows a possible relationship between INSPIRE articles and the policies proposed in section 2.3.6.

Table 2.8: A possible relationship between INSPIRE articles and the policies defined in table 2.5

Policy	**INSPIRE articles**
Governance policy	Art. 19: There is an INSPIRE coordinator at the European level, assisted by some organizations.
Role assignment policy	Art. 4 establishes which datasets and services are covered by the directive (i.e., which datasets and services will take the role of **spatial assets** in INSPIRE). Art. 19: The INSPIRE coordinator will be the European Commission. The Commission will also fulfill the role of **contact** in the interactions with Member State contact points. The European Environment Agency will be assisted by the INSPIRE coordinator. In different articles, public authorities are allowed to take different roles (**user**, **contributor**, **contact**). In different articles, third parties are allowed to take different roles (**user**, **contributor**).

Table 2.8: A possible relationship between INSPIRE articles and the policies defined in table 2.5

Policy	**INSPIRE articles**
Infrastructure policy	Art. 21: Member states must monitor their SDIs and report to the European Commission. Art. 23: The Commission will report on the implementation of INSPIRE to the European Parliament.
Standards policy	Art. 7: it requires to establish some implementation rules in order to address interoperability and harmonization issues in data sets and services. Art. 10: information needed for compliance with Art. 7 will be made available. Art. 12: Member States will ensure that public authorities and third parties are given the technical possibility to link their compliant services to those provided by them. Art. 16: rules for implementing network services as they have been established in the directive will be developed, including technical specifications. Art. 20: the implementation rules will take into consideration standards adopted by European and other International standardisation bodies.

Table 2.8: A possible relationship between INSPIRE articles and the policies defined in table 2.5

Policy	**INSPIRE articles**
Foundation policy	Art. 8: The implementation rules mentioned in Art. 7 will establish how to address certain harmonization aspects for the spatial data sets corresponding to the themes listed in Annexes I and II of the directive. These harmonized data sets, and some other elements listed in Art. 8 (unique identifiers framework, multilingual thesauri, consistency rules among spatial objects etc.) will be INSPIRE **core spatial assets**, so the part of the implementation rules addressing this will be part of the **foundation policy**. Art. 11 establishes several network services that must be provided by Member States to discover, view, download, transform and interconnect spatial data services.
Quality policy	Art. 5.3: Member States must ensure their metadata are of enough quality. Besides this, the Implementation Rules mentioned in Art. 16 may establish quality parameters on different components of INSPIRE (performance requirements for network services, mandatory fields for metadata etc.).
Promotion policy	There are not any activities that should obviously be considered promotional in the directive. Nevertheless, third parties may be involved as users, contributors etc., and Member States must be sure their public authorities benefit from INSPIRE: these activities may involve promotion.
Education policy	Art. 18 includes a mention to the necessity to provide information on existing practices to the different actors in their SDIs.

Table 2.8: A possible relationship between INSPIRE articles and the policies defined in table 2.5

Policy	INSPIRE articles
Funding policy	Art. 24 establishes the obligation for Member States to make the necessary laws, regulations and provisions to comply with INSPIRE: this includes some funds.
Access policy	Art. 5.2: Metadata will include information about conditions applying to access to spatial data sets and services. Art. 13: under some circumstances, Member States may limit public access to spatial data sets and services. Art. 14: some services must be free of charge, but some others may require some payment, provided there are some e-commerce service available for it. Art. 11: certain networks services will be established to allow for access to spatial data sets. Art. 15: Member States must make their services available through the INSPIRE geo-portal. Art. 17 indicates some general guidelines for the sharing and access to spatial data sets.
Membership policy	Art. 2: INSPIRE shall build upon the Member State SDIs. Art. 17 establishes some obligations for Member States in INSPIRE. Art. 18: Member States will coordinate the contribution of the different actors in their SDIs. Art. 19: Member States will designate contact points to work with the European Commission.

2.4.6 Interactions and Processes

As objectives, actors, artefacts, objects and policies in INSPIRE are compatible with those proposed in the pattern, i.e. similar concepts or refinements of them, we expect that it will be possible to model INSPIRE interactions and processes in a compatible way too. After all, interactions and processes are modelled for enterprise objects, that fulfill actor and artefact roles, to achieve objectives under the rules established by the policies.

An analysis of the INSPIRE directive allows us to find out references to behaviours that could be modelled as interactions, like those proposed in section 2.3.7. These interactions are shown in table 2.9.

Table 2.9: References in the INSPIRE directive to interactions compatible with those described in 2.3.7

Interaction	**Interaction in INSPIRE**
Join/Leave SDI	As Member States transpose INSPIRE to their laws, establishing their SDIs as required, they will join the INSPIRE SDI. These interaction, as defined in section 2.3.7, require certain components: • Several policies, described in section 2.4.5. • A **contact** initiating the interaction: A Member State contact point. • A **contact** responding to the interaction: The European Commission. • An **operational body** responding to the interaction: An assistant of the European Commission. Member States are also responsible for establishing their SDIs according to their own political organization. Therefore, they will have to establish join, and possibly leave, interactions, although this is not addressed in the directive.

Table 2.9: References in the INSPIRE directive to interactions compatible with those described in 2.3.7

Interaction	Interaction in INSPIRE
Contribute spatial asset	There are many references in INSPIRE to contributions of spatial data and spatial data services. These contributions, modelled as the interaction proposed in section 2.3.7, require certain components: • Several policies, described in section 2.4.5. • A **contributor** initiating the interaction: A public authority or a third party. • An **operational body** responding to the interaction: An assistant of the European Commission. • An **spatial asset** being contributed: Those datasets and services indicated in Art. 4. • An **spatial asset metadata** of the asset being contributed: The metadata described in Chapter II of the directive. • A **communication channel** and an *SDI catalog*: The INSPIRE geo-portal may fulfill both roles.
Withdraw spatial asset	When allowed, see for example Art. 13, Member states may withdraw, temporarily, a previously contributed asset. This interaction is similar to the **contribute spatial asset**. The only difference is that it does not require the **spatial asset** being withdrawn, only its metadata.

Table 2.9: References in the INSPIRE directive to interactions compatible with those described in 2.3.7

Interaction	Interaction in INSPIRE
Access spatial asset	This interaction is at the core of the objectives of any SDI, and INSPIRE is no exception. If modelled as the interaction proposed in section 2.3.7, it requires certain components: • Several policies, described in section 2.4.5. • A **user** initiating the interaction: users are not defined in INSPIRE, but they are taken into consideration many times in the directive. • An **operational body** responding to the interaction: An assistant of the European Commission. • A **contributor** responding to the interaction: The public authority or third party that originally contributed the asset being accessed may be required to access it. • A **communication channel** and an *SDI catalog*: The INSPIRE geo-portal may fulfill both roles. • An **spatial asset** being accessed: Those datasets and services indicated in Art. 4. • An **spatial asset metadata** of the asset being accessed: The metadata described in Chapter II of the directive.

Interactions and, especially, processes are in general too detailed to be described in a European directive. INSPIRE requires some implementation rules to detail some aspects, and surely these rules will address some processes, but these rules are still in development, and some of their results will be too detailed to be captured by a pattern as the one presented in this chapter. Metadata creation (Art. 6), information dissemination (Art. 10), network service establishment (Art. 11), monitoring infrastructures (Art. 21.1) or reporting (Art. 21.2) are some examples of behaviours included implicit or explicitly in the directive. A full architecture of INSPIRE should model these behaviours, either as processes or as interactions, but this is beyond the scope of this work.

2.5 Conclusions

This chapter has presented an architectural pattern for SDIs that allows to model them as federations of communities in terms of the enterprise language of the ISO RM-ODP. This pattern proposes a systematic approach to model the purpose, expected behaviour and policies of an SDI, starting from the communities that compose it and the relationships among them. These relationships may be hierarchical, but other arrangements are allowed if needed. The pattern includes a minimum set of SDI components commonly found in the bibliography, and others that have been found necessary under the guidelines provided by the enterprise language of the RM-ODP.

Architects and designers may apply this pattern as it is to produce an enterprise architectural view of a given SDI, or they may extend and refine some of its elements if needed. In any case, the use of this pattern provides a starting point founded on solid concepts, provided by the RM-ODP, and a way to facilitate the exchange of knowledge about SDI models.

To test the applicability of this pattern to a complex SDI, and also as an example of use, the INSPIRE directive has been analyzed. It has been shown that most components and rules addressed by INSPIRE are similar to those proposed in the pattern, or are specializations that can be modelled extending them.

IIs have been pointed out as a conceptual base for SDIs by several authors. One of the implications of that is that SDIs cannot be fully designed and then implemented. SDIs are complex systems in constant change, with many different components, and where many actors, with different interests, necessities and degrees of autonomy, are constantly interacting. We thus expect that the pattern proposed in this work can be useful to model SDIs to a certain extent (i.e. fundamental roles, policies, interactions etc.), that is enough to set up an 'SDI game board' where the evolving interactions among its communities, actors, objects etc. can take place, an be followed. Further research is needed to validate this point, and also to evaluate other conceptual frameworks, especially systems of systems and federations of systems [152], which can prove themselves useful to tackle the complexity that is inherent to SDIs.

Chapter 3

A Component & Connector Architectural Style for Spatial Data Infrastructures

3.1 Introduction

From different points of view, Spatial Data Infrastructures (SDIs) have some characteristics of digital libraries [138] and of Information Infrastructures (IIs) [74]. From the II point of view, an SDI may be formed by several interconnected systems that could be seen as SDIs themselves. When one of these systems is responsibility of one organization, it is usually not considered as the piece of an infrastructure, but as an information system which must adhere to certain rules and principles, to facilitate its interaction with others, and which has its own requirements and organization. This aspect of SDIs must be taken into account when addressing its design, creation, and maintenance and it is the main focus of this section.

The original definitions of SDIs already include, directly or indirectly, the necessity to provide search, visualization and data download services [78]. Nevertheless the current research trends in the evolution of distributed interoperable Geographic Information Systems (GIS), based on standard [83, 16] and semantically enabled [104, 112], geographic Web services [43], have made some authors argue that SDIs will soon include all kind of geographic Web services [14, 177]. This trend will produce SDIs more and more complex. One way to cope with this complexity is to have the appropriate mechanisms to describe these infrastructures in architectural terms.

There are several previous works which propose reference and/or architectural models for SDIs. Among the most relevant ones it is worth mentioning the description of the USA National SDI [58]; the proposal in [70] about the Canadian infrastructure and the initial proposals for the European SDI established in [91]. All these architectural models share the same problems:

- They are focused on the allowed components for the architecture and they barely mention, when they are mentioned, other elements of an architecture, as the types of relationships among the components, their visible properties or necessary constraints.
- There are non-obvious overlappings among the different architectural models. Components with different names but similar roles are common.
- They are not completely grounded in well-known architectural models. A Service Oriented Architecture (SOA) [53] is commonly mentioned as the basis, but this architectural model fails to capture many components included in these models like applications or data repositories; no other references to software architecture models are provided.

As the documented reference models for SDI architectures in the bibliography do not follow any kind of common structure or pattern, comparing them, or verifying that a GIS follows one of these architecture reference models, is difficult and ad hoc. In this chapter, an SDI architectural style is defined to capture the current knowledge about SDI architectural models while avoiding the problems of the currently published models.

The rest of this chapter is structured as follows: Firstly an architectural style for SDI is defined, after analyzing the architectures proposed by six SDI and geographic services reference models, following the 'Views and Beyond' proposal [45]. In section 3.3 three real SDI projects with published architectural descriptions are studied to determine their compatibility with the proposed style. The next section offers an architectural description of one of these projects, following the proposed style, to highlight the benefits of its use. In the final section some important issues are highlighted and conclusions are drawn.

3.2 An SDI style for the C&C viewtype

This section defines an SDI architectural style following the 'Views and Beyond' methodology, and thus under the umbrella of one of the viewtypes defined in [45]: the Component-and-Connector (C&C) viewtype. C&C views include elements with runtime presence, such as clients or servers, which are the components, and the pathways for their interactions, such as information flows, captured as connectors. A general C&C viewtype thus consists of allowed component and connector types, constraints for allowed relations (i.e. which connectors are attached to which components), some properties of the components and the connectors (i.e. a name and a type) and maybe also some topological constraints.

The objective behind defining this style is to capture, unify and systematize the previous knowledge on SDI architectural models, and to explicitly take into account elements that are not typically considered in these models (i.e. constraints), or considered only implicitly (i.e. data stores). Thus on the one hand, this architectural style must provide:

- A tool and a shared vocabulary to help system architects to design SDIs, bringing to light those elements that a system architect must consider when putting together an SDI.
- A method to document relevant facts of an SDI architecture, considering as relevant facts those which are present in most SDI architecture proposals.

But on the other hand, the style does not need to provide:

- A method to completely document the architecture of an SDI. As described in section 1.4, several views following different viewtypes and styles would be needed to achieve this.
- A one-size-fits-all solution for SDIs: it must be a pattern extracted from SDI reference models and from SDI projects, which brings to light common elements,

and important missing elements, and gives them a structure, but it does not intend to be a closed and fixed architecture for SDIs.

A *hybrid style* is defined in [45] (p. 201) as the combination of two or more existing styles. From the styles for the C&C viewtype in this book, those that have been considered more appropriate as a basis for this work are Shared-Data and Client-Server. The proposal in this chapter is a specialization of a hybrid style which combines these two:

- **Shared-Data**: This style highlights interactions dominated by the exchange of persistent data. It is important for SDIs because spatial data sets and metadata are persistent data, shared by different kinds of services and very relevant. In this style, there are two types of components: *shared-data repositories* and *data accessors*. The possible connector types are *data reading* and *data writing*. *Data accessors* are attached to *data repositories* by means of these types of connectors.

- **Client-Server**: This style shows asymmetric interaction among components, from clients to servers. It is important in SDIs because they follow an SOA: some of their services will act as servers, for other services or for applications, and others will act as clients for other services, and these interactions are the base to develop complex functionality. In this style, there are also two types of components: *clients*, which request services and *servers*, which provide them. The connector type is thus *request/reply*. *Clients* are attached to *servers*.

The next sections describe the elements of a new style for SDI. These elements extend those in the Shared-Data and Client-Server styles to tailor them to the necessities of a software architect designing an SDI. This style has been designed from the experience of the authors in several SDI projects [24, 23, 109, 143] and taking into consideration several of the most relevant SDI and geoservices architecture descriptions in the bibliography. A discussion about the relationship between the elements proposed here and those in the bibliography is also presented.

3.2.1 Previous work on SDI architectural models

These are the main bibliographic references that have been taken into consideration, and the reasons to choose them:

- The ISO Technical Committee 211 (ISO/TC 211) standard on geographic information services (ISO 19119, [92]): this is the most thorough taxonomy of geoservices available. From a technological point of view it is an abstract specification, but most, if not all, current SDI initiatives are using Web services and this technology fits very well with the ISO standard.

- The OGC Web services architecture description [174]: the geoservices architecture from the most active standardization organization, with ISO, in the geospatial field. It is quite similar to the ISO standard, but it is technologically specific (Web services, based on Web protocols or SOAP, and XML to transfer data).

- The USA Federal Geographic Data Committee (FGDC) Geospatial Interoperability Reference Model (GIRM) [58]: the concept of national SDI was developed in the USA, and the FGDC set up this guide, one of the first and most relevant for these kind of infrastructures. Besides this, this model was included in the first position paper on architecture for the Infrastructure for Spatial Information in Europe (INSPIRE) [12].

- The Canadian Geospatial Data Infrastructure Architecture Description [70]: the architecture of one of the leading projects in national SDIs in the world.

- The final text of the European Union Directive for the establishment of a European SDI (INSPIRE, [57]): relevant because it establishes the minimum requisites for all national SDIs of the EU member states to be part of a European SDI. Although it could be considered that it does not define an architecture, the truth is that although there are not any diagrams, it gives some detail on the components that national SDIs in the EU must have, in some cases more deeply than other architectural proposals.

- A proposal from the European Commission (EU) Joint Research Center (JRC), presenting the initial steps leading to the establishment of the European Geographic Information Portal [17]. The JRC is the institution in charge of providing scientific and technical support of the EU policies, among them INSPIRE.

Although of course this list may never be complete, a reference to the Global SDI (GSDI) could be expected. But the GSDI cookbook [78] does not suggest an SDI architecture reference model; it refers to other documents for this (specially ISO and OGC standards) which have been considered.

3.2.2 Component types

The component types in this SDI style are specializations of those in the client-server and shared-data styles defined in [45]. They have been designed from the experience in several SDI projects and from the bibliography on SDI and geoservice architectures, as explained before. Regarding to this, although the ISO 19119 standard is platform-neutral, most other bibliography on SDI and geoservices assumes a Service Oriented Architecture [53], deployed over Internet protocols with XML as the data exchange format, i.e. Web Services [31]; this is also the case of this work. The component types have been chosen because they play relevant roles in SDIs, but not all of them need to be present in every SDI. Their names have been selected from the bibliography when there seemed to be a high degree of consensus. When this has not been possible, they have been chosen to highlight their main characteristic. The intention has been to capture the main structure of an SDI, so the component hierarchy is not very deep. ISO and the OGC have done a good work specifying types of geoservices, so in this chapter only the higher levels in the component hierarchy, those which hold a higher level of information about the structure of an SDI, have been defined.

Figure 3.1 shows the hierarchical relationships among these component types, and among those in the client-server and shared-data styles. This is a Unified Modeling Language (UML) class diagram where classes represent component types. Table 3.1 holds a comparison of these component types with those that appear in the considered bibliography. The table shows which of the proposed component types appear in the different architectures studied. It also indicates when they appear with a different name, with a similar, but not equal, meaning, or when they do not appear but are related, even indirectly, to other explicit elements. The definitions of the proposed component types are given in the next list:

- **Web Service**: All kind of Web services [31].
- **SDI Service**: All Web services in an SDI will be a specialization of an SDI Service. The name has been chosen to reinforce the idea behind the architectural style while avoiding other names than could be understood as too restrictive (i.e. calling them geographic information services or geoservices seems to imply that they all access geographic data, and this will not be the case for some of them).
- **Processing Service**: These services are designed to make generic processing of data, typically spatial data. These data can be provided when calling their operations, or the services can access to some data repositories.
  - **Transformation Service**: Services that allow spatial datasets to be transformed, with a view to achieving interoperability.
- **Information Management Service**: They store and provide access to data and metadata.
  - **Portrayal Service**: They support the visualization of spatial datasets.
  - **Access Service**: These services allow to download, or access, spatial data sets or parts of them.
  - **Catalog Service**: These services make it possible to discover, explore and evaluate datasets, services etc., by means of the metadata describing them.
  - **Gazetteer Service**: They offer geocoding functionalities, which link toponyms and their spatial location.
  - **Knowledge Model Service**: They offer discovery and access to shared knowledge models in order to facilitate the semantic interoperability among different services, applications etc.

- **Application Service**: Those used to support client applications, specially thin, i.e. Web, clients.
- **SDI Client**: Software that gives human users access to the services in an SDI.
  - **Application**: A computer software that allows users to perform a set of tasks, most of them using SDI Services.
  - **Geoportal**: Web sites mainly focused on geographic content, geographic services, and the tools to discover them. Although it would possible to model a Geoportal as a type of Application, the relevance of Geoportals for SDI in the bibliography supports considering them on their own.
- **Metadata Repository**: A repository which holds metadata, being metadata identifiable and structured data about other resources in the SDI (datasets, services etc.).
- **Knowledge Model Repository**: A repository which holds knowledge models, defining knowledge models as data models, schemas, ontologies, thesauri or any other explicit conceptualization of knowledge in a domain.
- **Dataset Repository**: A repository that holds datasets, defining datasets as identifiable collections of data.
  - **Spatial Dataset Repository**: A repository that holds spatial datasets, which are defined as identifiable collections of spatial data (i.e. data with a direct or indirect reference to a specific location).

3.2.3 Connector types

In the bibliographic references on SDI and geoservices architecture listed before, there is little attention to connectors. At most there are some indications about what kind of components can connect to others, without further details. This could be due to the fact that defining special connector types seems not necessary for SDIs; but since connectors in general are barely mentioned it could also be possible that they have

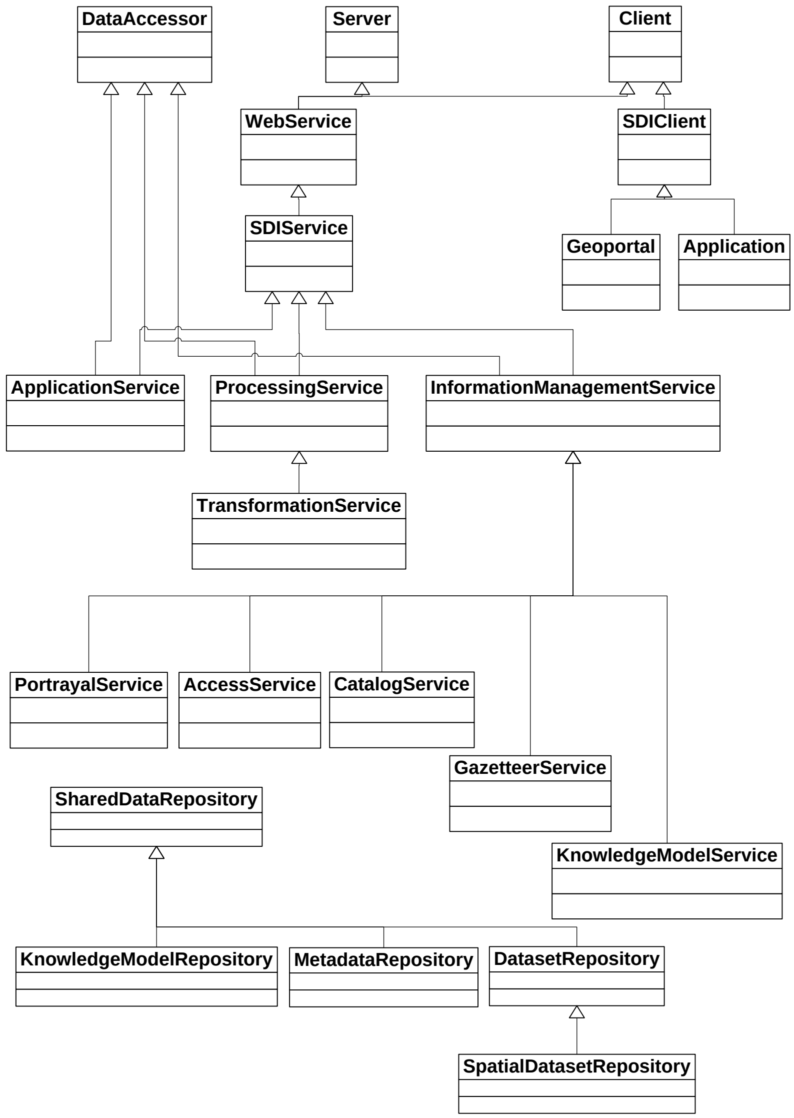

Figure 3.1: Hierarchical relationships among SDI style component types

Component Type	OGC	FGDC GIRM	ISO 19119	INSPIRE	EU Geoportal	Canadian GDI
Web Service	✓	✓	×	✓	✓	✓
SDI Service	✓ (OGC Web Service)	×	✓ (Geographic service)	✓ (Spatial data service)	✓ (GI Service)	✓ (Service)
Processing Service	✓	✓ (Geoprocessing service)	✓ (Geographic processing service, Geographic communication service)	×	✓	×
Transformation Service	×	×	×	✓	×	×
Information Management Service	✓	×	✓ (Geographic information/model management servie)	×	×	×
Portrayal Service	✓ (Web map service)	✓ (Maps & visualization service)	✓ (Map access service)	✓ (View service)	✓	✓ (Rendered map service)
Access Service	✓ (Web feature service, Web coverage service)	✓ (Web feature service, Web coverage service)	✓ (Feature access service, Coverage access service, Product access service)	✓ (Download service)	✓ (Feature service)	✓ (Geographic features service, geographic feature encoding service)
Catalog Service	✓ (Catalog service for Web)	✓	✓ (Catalogue service) (1)	✓ (Discovery Service)	✓	✓ (Registry)
Gazetteer Service	✓ (Web gazetteer service)	✓	✓	× (2)	✓	×
Knowledge Model Service	× (10)	×	✓ (3) (Registry service)	×	✓ (Thesaurus Service) (9)	× (8)
Application Service	✓	×	✓ (Geographic workflow/task management service)	×	×	×
SDI Client	✓	✓ (4)	✓ (Geographic human interaction service)	×	×	×
Geoportal	× (5)	✓ (Geospatial portal)	×	✓ (Geo-portal)	✓	×
Application	×	✓ (User application)	×	×	× (6)	✓
Metadata Repository	× (7)	× (7)	× (7)	× (7)	× (7)	× (7)
Dataset Repository	× (7)	✓ (Content Repository)	× (7)	× (7)	× (7)	× (7)
Spatial Dataset Repository	× (7)	✓ (Feature and Coverage Repositories)	× (7)	× (7)	× (7)	× (7)
Knowledge Model Repository	× (7)	× (7)	× (7)	× (7)	× (7)	×

×: This architecture does not include explicitly this component type
✓: This architecture includes this component type with similar semantics, though maybe with a different name (it is given within parentheses).

(1) There is also a Registry Service in the ISO 19119 standard, but with a different meaning.
(2) Not mentioned explicitly in the INSPIRE directive, but an indirect reference seems implied when it indicates that the discovery services will support search by geographical locations.
(3) The ISO 19119 registry service provides access to metadata about types, so it could be considered a specialization of the knowledge model service.
(4) The FGDC GIRM mentions clients in its interoperability stack, indicating that user applications will act as clients in the distributed system they propose.
(5) OGC defines Web portal services, but they are application services as defined in this work.
(6) The geoportal is defined in this work as an application that aggregates instances of GI services, but application is not defined and not mentioned any more.
(7) Not explicit components, but they are indirectly taken into consideration (i.e. considering [some of] the contents of the store type).
(8) The architecture does not reference it, but a geospatial ontology server, a specialization of the knowledge model service, has been created for this SDI (M3GO, http://intelecgeomatics.com/ogm3/).
(9) A thesaurus service as defined for the EU Geoportal could be considered a specialization of a knowledge model service.
(10) The catalog in the OGC architecture can store metadata and datasets which can be schemas, models, semantic documents, etc., so it could play the role of the knowledge model service.

Table 3.1: SDI style component types compared with those in the other architectures studied

not been considered at all. After a more detailed study, neither new relevant types of connectors nor refinements of those provided by the Client-Server and Shared-Data styles have been found. Therefore these are those included in the SDI style:

- From the Client-Server style:
  - **Request/Reply**: The invocation of a server by a client and its response go through this connector type. In the SDI style, SDI Clients (i.e. Applications) or SDI Services can make requests to SDI Services, and the latter can reply to the former.
- From the Shared-Data style:
  - **Data Reading**: Data accessors read data from data repositories. In the SDI style different types of SDI Services can read data (i.e. Information Management Services).
  - **Data Writing**: Data accessors write data to data repositories. In the SDI style different types of SDI Services can read data (i.e. Access Services).

3.2.4 Properties

As with the connector types (see subsection 3.2.3), it has not been possible to find relevant properties for SDIs that were present in a significant number of architectural proposals, and missing in the generic C&C styles (Client-Server and Shared-Data). But among the several properties for these styles suggested in [45], there are some that are used in some studied SDI architectures:

- **Name**: for components and connectors, suggesting their functionality or the nature of its interactions.
- **Type**: the type which components and connectors belong to.
- **Types of data stored**: for **Shared Data Repositories**.

This list of properties is not closed. System architects may consider useful adding others when designing their SDIs following the proposed style. For example properties indicating access permissions or performance indicators could be useful. They just seem a little too specific for the objectives of this work.

3.2.5 Constraints

As defined before, constraints in an architectural style are rules which specify how the elements defined for the style, specially components and connectors, can be used, and the valid interactions among them. This section defines some fundamental topological constraints, which are those that define how components relate to each other by means of connectors.

First of all, these are the allowed connector configurations (topological constraints) defined for the Client-Server and Shared-Data styles (they have already been mentioned when describing the Connector Types):

- From the Client-Server style:
  - Client **Requests** from Server.
  - Server **Replies** to Client.
- From the Shared-Data style:
  - Data Accessor **Reads Data** from Shared-Data Repository.
  - Data Accessor **Writes Data** to Shared-Data Repository.

In the studied SDI and geoservices architectures and models, there are not many clear references to constraints. Nevertheless some can be found:

- **OGC**: This architecture describes some ideas which are constraints indeed. They would be more clear if they were separated and made explicit. The constraints designed for the style in this chapter are compatible with these ideas.
  - Services are organized into tiers but loosely, and it is not required to separate services that way. Services can use other services within the same tier or not.

  - All kind of services may access data, although most of data will be accessed by Information Management Services.

- **FGDC GIRM**: This model organizes its components in an 'interoperability stack'. In this stack, user applications have access to services and to content repositories (direct data access), and services access other services and content repositories. In our proposal **Clients** are not **Data Accessors**, so they are not allowed to read or write to **Shared Data Repositories**; this is more restrictive than the GIRM proposal, where applications can directly access content repositories. As most other SDI proposals separate clients from data by means of services, this constraint has been included in the style designed in this chapter.

- **ISO 19119**: In this standard, the engineering viewpoint section establishes as a reference model a 4-tier logical architecture. This logical architecture is then mapped to different physical ones, establishing thus some constraints on the topology of interactions among services. The problem is that this architecture is designed for generic Information Technology (IT) services as well as for GIS-extended services, so it is a general proposal with a broad scope. If besides this we consider that this standard is not for SDIs but for geoservices in general, it results that the level of detail is not appropriate to extract conclusions useful for an SDI style as the one defined in this work.

When defining the SDI style, new component types have been pointed out. These component types extend those in the Client-Server and Shared-Data styles, so they inherit their constraints too. But not every component type extending Data Accessor should be allowed to read *and* write from/to *any kind* of Shared-Data Repository. New constraints are needed to explicitly capture these new rules. These constraints are given as forbidden topological connections among some component types:

- Portrayal Service:

  - **NOT Writes to** Shared Data Repository.

  - **NOT Reads from** Knowledge Model Repository, Metadata Repository.

- Access Service:
  - **NOT Reads from AND NOT Writes to** Knowledge Model Repository, Metadata Repository.
- Catalog Service:
  - **NOT Reads from AND NOT Writes to** Knowledge Model Repository, Dataset Repository.
- Gazetteer Service:
  - **NOT Reads from AND NOT Writes to** Knowledge Model Repository, Metadata Repository.
- Knowledge Model Service:
  - **NOT Reads from AND NOT Writes to** Metadata Repository, Dataset Repository.

It is important for a better understanding to clarify some points, and to highlight a few consequences of these constraints:

- These constraints intend to separate the roles of the different service types. For example, a Portrayal Service is specifically designed to portray existent spatial datasets so, although it is basic to allow it to read Spatial Dataset Repositories, it is not allowed to modify them or to read from other types of repositories. If one of these services would be needed to read a knowledge model or some metadata, and this situation is perfectly possible, it should do it through a specialized service (a Knowledge Model Service or a Catalog). This is in order to follow good design principles, like a clean and strict separation of service roles. But in some situations these constraints may be unnecessarily complex: for example one could want to create a Catalog Service able to read just some data from a Spatial Dataset Repository, but without the burden of setting up an Access Service. This can be done by defining a new component type which extends Catalog Service *and*

Access Service. This component would be thus allowed to read from a metadata repository *and* from a dataset repository. The idea behind constraining the data repositories which can be accessed from different components is to help to clarify their function; but a system architect may decide that for a specific SDI a catalog component which accesses metadata and datasets is the best solution. This style allows for that while making it explicit that this component is a Catalog Service *and* an Access Service. Making it explicit is useful because it gives roles and precise meanings to the elements in an SDI, and because it helps this system architect to document the design, relating this component to the component types defined for this style. This also makes the design easier to understand to other system architects who know the SDI style.

- Geoportals and Applications are not allowed to access Shared Data Repositories, because they are not Data Accessors. If this necessity arises in the process of designing an SDI, it is a clear indication that some Application Services are needed. This is one of the reasons why Application Services have been defined: to separate Applications from the Dataset Repositories, helping to enforce the usual rules of layered IT systems.

- Portrayal Services have been allowed to read from Dataset Repositories. It could be argued that this would be the role of an Access Service and that most Portrayal Services in SDIs would have to be also Access Services. This decision has been taken precisely because the main function of Portrayal Services is reading Spatial Datasets and portraying them. If their main function includes reading Datasets, it seems correct to allow them to read from Dataset Repositories.

As in the case of property types (subsection 3.2.4), this list of constraints is not closed. They have been chosen because they capture the basic ideas which appear, normally in an implicit manner, in the SDIs studied, and have proven themselves to be useful in the experience of the authors with SDI projects.

3.3 Analysis of real SDI architectures

In this section three different projects are analyzed in order to determine if real SDIs have architectures that fit the proposed style. These projects are from regions in three different European countries and have been developed by different people, with different technologies, objectives and constraints. They have been chosen because they give enough public architectural information, claim to be following SDI principles and have a view that is close to the C&C viewtype. The Galicia CMA SDI project can be analyzed in more detail because of the implication of the authors of this work in its development.

3.3.1 Architecture of the Galicia CMA SDI

Plans for the adoption of INSPIRE framework legislation have been encouraging EU national and regional governments to start giving effective steps towards the creation of the geographic data and services infrastructures this legislation is establishing. This is possible now because the relevant standards and architectures that will be adopted have already been proposed, and there already exist implementations for many of the different components needed. This is the case of the Galicia Department of the Environment (Consellería de Medio Ambiente, Xunta de Galicia, CMA). This department had found the same kind of problems with geographic information that INSPIRE addresses: incompatible data formats and information systems, difficulties disseminating data among their users (it is a very decentralized department), difficulties to find relevant information, etc. The solution adopted to overcome these problems has been to develop a geographic information system for this department, following INSPIRE principles and recommendations in architecture and standards, thus effectively building an SDI. This infrastructure has been designed to become the core of a future Galician SDI.

Galicia is located at the northwest corner of the Iberian Peninsula. The climate is warm and wet so its land is covered with many forests (69 percent of its surface). This fact makes forests the main concern of the CMA, with water use, disposal of waste and protected natural environments among its other responsibilities.

Galicia is divided into four provinces, but the CMA divides it also into nineteen forest districts, in order to address the necessities (reforestation, forest fires, cleaning...) of such a big forest surface. This is the main reason behind the decentralization of this department, with only 10 percent of its workers at the central building in Santiago de Compostela, and the rest of them at the provincial delegations and the forest districts. Currently this department is finishing the wiring of all provincial delegations and forest districts. Once the wiring is complete, all delegations and districts will be connected to the department Intranet. The decentralization of the CMA makes the usual problems with geographic information in big organizations and public administrations much worse. It is difficult for users to find the data they need or even to find out if that information exists. In some cases, i.e. people in forest districts away from the central building, users just did not have any access to geographic information that would make their work much easier. Another problem is the delay in the building of an integrated solution that has led some districts to adopt different GIS software solutions, or no solutions at all in some of them. This situation has added problems of data formats and interoperability.

User requirements

Given the problems related to geographic information management and use in the CMA, and the development of the INSPIRE initiative, with the requirements it will impose to EU members in some years, building an SDI following this initiative principles was the best option to address both issues simultaneously. This would solve the CMA geographic information users' needs while giving some effective steps in order to fulfil the future INSPIRE legislation, making profit of the recent networking of all delegations and districts.

The first work was, of course, collecting users' requirements. Here is an overview of the most important ones:

- All geographic information, raster and vector coverages, must be stored in a spatial database.
- It will be possible to find relevant geographic information available.

- Users will be given tools to view the geographical information they need over the CMA Intranet. They will also be given the possibility to make spatial and non-spatial queries to this information.

- Users will also be given the possibility to download the geographical data (vector and raster coverages) in the format they are used to work in.

- Advanced users will be able to access simple map services, designed to be combinable with their local data, and to query the information viewed in this map services.

- Software already owned by the CMA should be used where possible. This includes: ESRI ArcSDE 8.x and ArcIMS 4.x, Oracle 8i/9i with the spatial cartridge, the Safe Software universal translator FME, and its web version Spatial Direct.

Other requirements where specified and recommended to the CMA, after studying their necessities and the guidelines offered by INSPIRE and the GSDI, briefly:

- Standards must be followed, if possible and where available, specifically those standards recommended by INSPIRE and the GSDI.

- Architecture will follow that recommended by INSPIRE. There will be data, a metadata catalog, and chainable services offering at least visualization, access and data searches.

All this needed to be accomplished in a short period of time (half a year), and there had to be visible results in half that time, in order to convince the decision makers to approve the needed funding for improving and expanding this infrastructure. This short deadline obviously conditioned some parts of this work.

As one of the requirements was to use the already available software in the CMA, a COTS (commercial off-the-shelf, see for example [3]) based approach was decided. This provided both advantages:

- Most GIS software offers capabilities that have already been tested by both the vendor and the market. A COTS architecture allows for making profit of these complex capabilities by integrating this software as a component [3].

- The commercial software licences in the CMA (products such as ESRI ArcIMS, ArcMap, ArcInfo and ArcSDE, Oracle 9i and SAFE FME and Spatial Direct) are of complex, and expensive, products. These fulfil, at least partially, requirements of this infrastructure, and should thus be used for a faster development.

These are the software products and components that where integrated for the project:

- All the geodata (both vector and raster data) and metadata used in this project are stored in an Oracle 8i object-relational database with the spatial cartridge (Oracle Spatial) to provide geographic information support (spatial storage format built around OpenGIS Simple Features specification, spatial queries, spatial indexes, etc.).

- ArcSDE, that is an ESRI gateway that facilitates managing spatial data in different database management systems. It has not been used here to access different databases or to take advantage of the spatial capabilities it provides, since all the data is stored in Oracle Spatial, but as a middleware component to provide good compatibility with ESRI products, including of course ArcIMS and all the other ESRI applications in use in the CMA, while keeping all the data stored in Oracle. ArcSDE is used also to give an entry point for other non-ESRI components that need to access the data, such as Spatial Direct or a developed WCS wrapper. Although it has powerful spatial capabilities, in this architecture ArcSDE makes use of the spatial management facilities provided by Oracle in order to facilitate data access by other software products or components that may be incorporated into the infrastructure in the future. While ESRI applications can access directly the data through ArcSDE (and that is the case of the ESRI applications that are installed inside the CMA, such as ArcMap or its previous versions, ArcInfo, ArcView and ArcCatalog), other products, such as Intergraph GeoMedia, can access much more easily to the data directly through Oracle.

- ArcIMS 4, ESRI Internet Map Server, which is a software product that can produce representations (images) and deliver content (vector data) of maps through the Web. Since it is compatible with the OGC WMS 1.0 specification and with the WFS 1.0 specification, it has been used to provide the needed OGC standard interfaces.

- A metadata catalog implements a set of service interfaces that support management, discovery, and access of geospatial information, following the OGC Catalog Services specification. This component has been developed from scratch by our research laboratory at the University of Zaragoza [135], but not specifically for this project, so, in spite of not being a commercial product, it follows the COTS philosophy.

- Vector data could be provided in a standard way by the use of a Web Feature Server. While the WFS is not integrated into the infrastructure yet, and because the user will usually need the data in some specific format, Spatial Direct 2002, a SAFE software component for the Internet download of vector data is used here in combination with FME, a component for geodata format transformation to allow the download of vector data in a variety of formats and spatial reference systems. Although Spatial Direct is not based on standard interfaces, its functionality can be used to perform this task and the component can be easily integrated in the system.

- A Web Coverage Server wrapper had to be built on top of ArcSDE to provide raster data. As at the moment the infrastructure was built there were no commercial products compliant with the OpenGIS WCS specification and given the internal capabilities of ArcSDE for managing raster coverages, a Java servlet access the ArcSDE functionality through its C language API (since the Java API is not completely implemented) and offers access to raster data through a subset of the too much complex interfaces of the OGC WCS standard specification. This

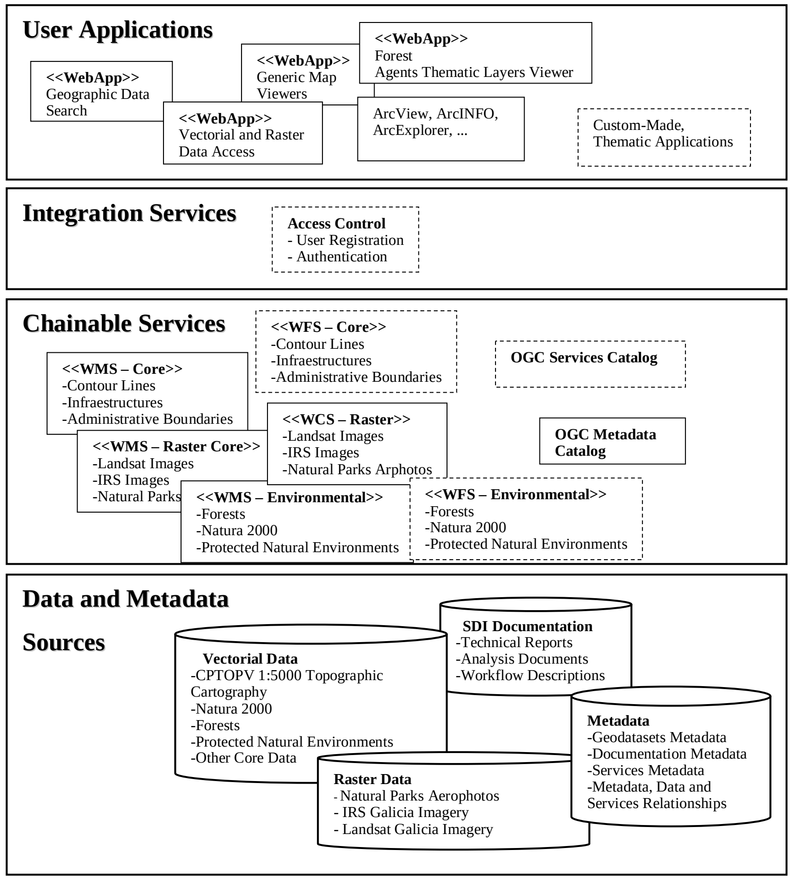

Figure 3.2: Galicia CMA SDI Architecture (taken from [24])

is the only component, apart from the final user applications, that has been built from scratch and specifically for this project.

Architecture

Figure 3.2 shows a service-centered view of the CMA SDI. This helps to emphasize several important things that are the core of any SDI:

- The main components of an SDI are chainable web services. Through the set-up of standard OGC web services, syntactic interoperability is easily achieved.

This allows for an easier integration of different services, both coming from thIS infrastructure and from any other place, to quickly develop new custom-made or thematic applications in answer to the evolving users' needs.

- It also emphasizes that chainable services are built on top of geodata and metadata. This allows for a better semantic interoperability, as geodata are described by its associated metadata. Including metadata along with reference and thematic data is also a INSPIRE recommendation.

- User applications are built on top of distributed services, both chainable standard services and integration services. Integrated services would be those provided to support extended functionalities, i.e. not covered by the standard ones, not logically related to individual end-user applications.

After studying the users' needs and the available geodata (more on data in the next section), a number of visualization and access web services were planned (OGC standards: Web Map Service (WMS), Web Feature Service (WFS) and Web Coverage Service (WCS)), trying to:

- Allow for both visualization and access of all vector and raster data.

- Allow for an easily customizable integration of different data.

- Keep a controllable number of web services, aiming at an easier maintenance.

- Group related things together, in order to make it more intuitive.

Three simple conceptual categories were decided for the data (core vector (reference data, basic topographic information), environmental vector (thematic data related to the CMA) and raster) and web services were designed consequently. Besides visualization and access, the other main required functionalities were data searches and metadata management. The OGC catalog service covers these areas, so one metadata catalog was included.

In order to provide their users with some fundamental core geographic information, the CMA reached an agreement with the Department of Public Works, Land

Administration and Housing (its Galician acronym is CPTOPV) to use the 1:5000 Galician cartography they produce, which includes topographic information, administrative boundaries, communication infrastructures, public service infrastructures and some other layers. This cartography is distributed as approximately 4000 files. These files contain nine thematic layers, and are more CAD-oriented than GIS-oriented (i.e. contour lines are broken to accommodate labels), what makes it quite difficult to extract features from them. The CMA negotiated with other public entities in Galicia (as well as with private companies) to get Landsat and IRS images covering all their territory, in order to provide their users with raster data, at different resolutions. In the INSPIRE spirit, buying this information should not have been necessary, because providing access to this information should have been the responsibility of other departments in Galicia government, but this was not the case. The CMA is the first SDI initiative in Galicia, and needing to give their users a useful, and as complete as possible, service, there has been to take some decisions of this kind. Besides the information already mentioned, the environmental information owned or created by the CMA and made available through this infrastructure includes coverages, mainly shapefiles, of forest management (most of the data), Natura 2000, and protected natural environments (including some high resolution ortophotographs of several natural parks).

One of the user requirements was storing all these data in a spatial database. The next section describes the process of storing both vector and raster data in the database.

Data processing and storage

The objective was to use the universal translator software FME to store all vector coverages in Oracle Spatial format, although through ESRI SDE in order to allow for an easy access both from ESRI software and for other software directly compatible with Oracle Spatial. This process, briefly described, consists of designing FME filters (semantic and syntactic mapping between data in one format and data in other format) and applying them to all the data files (coverages) to be translated. Carrying out this has taken much more time than initially planned, and this fact should be taken into consideration in any similar project: almost 12 man-months have been needed, when initial plans where of 4 to 6 man months, at most, for vector data storage. The main problems found, and the solutions applied, were:

- Poorly specified data: The CPTOPV 1:5000 cartography, the bulk of the data to be stored in Oracle, was very poorly specified. The aim was splitting the original thematic layers into features, in order to allow for a better manipulation of them. In order to achieve this, a good knowledge of this cartography was needed. After recovering all information (i.e. some metadata) available on these coverages, and interviewing people from the CPTOPV, it was evident there was little control over the data: different companies had made it on different years, and there had not been an as good as possible quality assurance process. As a result there was a lack of reliable information that led to a trial-and-error, slow, process of extracting features from these coverages. Extracted features were then visually matched against the original files to achieve a minimum quality check.

- Geometry errors: False line segments (segments shorter than a small threshold), self-intersecting polygon boundaries, self-intersecting contour lines and other similar problems were found in the CPTOPV cartography, and in other data files. These errors prevented FME from storing these coverages correctly in Oracle and/or prevented Oracle from creating spatial indexes on them (very important for an efficient access). These kinds of problems made it difficult to create reliable batch processes to store coverages massively. The solution was making more robust FME filters, by including more geometry tests, to cope with these problems before starting the translations. Some other errors were reported to the creators / maintainers of the coverages in order to get them fixed.

Storing the available raster data, satellite and aerial photographs at different resolutions, in SDE / Oracle needed a different approach because FME does not support raster data. This was not really a problem because their treatment did not present semantic interoperability problems. Nevertheless other issues arise with raster data:

- Mosaicing: Satellite and aerial images of large areas come typically in smaller pieces (tiles). These tiles may be regular or not, and may present overlapping, white frames around them and a few other problems. Available images, GeoTIFF files, to be included in the CMA SDI presented a variety of these issues:

- The Landsat image covering the entire region was only one file.
- The available IRS imagery came in regular tiles, perfectly rectified and georeferenced.
- Aerial 1 meter resolution ortophotographs of natural parks came in irregular tiles, with overlapping and contrast / brightness / shadows differences but without frames or other major problems. Not having adequate software for airphoto mosaicing, a more creative solution had to be found in order to cope with the natural parks ortophotographs. An approach based on ArcObjects, the software components offered by ESRI Arc products, was finally decided. A small visual basic script was written in order to make profit from ArcObjects capabilities to create mosaics from different georeferenced images, even those presenting overlapping, and to store them in a D.B. through SDE. With a higher budget and more time, a more sophisticated approach could have been taken in order to color balance the aerial photographs etc., but the solution adopted was enough to cover the users' needs.

- Fast access: One of the main concerns with large images is allowing for a fast visualization of them at all resolutions. The typical approach to this problem consists of creating pyramids of images composed by the original one and several versions of it, all adequately partitioned and at different resolutions. This way, only the pieces of the images at the resolution actually needed by the user are loaded, vastly improving the efficiency. SDE can automatically create these pyramids for stored raster data, so this issue was quite easily solved.

Metadata creation process

Creating metadata about, or cataloguing, all this spatial information had two major objectives. The fist one was to organize and maintain the CMA investment in data, preventing data value being lost by losing the knowledge that makes data really useful, and encouraging its appropriate use. The second objective was to provide information to data catalogs, in order to allow for data searches and, in the long term, to facilitate the integration of this SDI with others, following the INSPIRE recommendations on distributed interoperable services, which include distributed catalog services. The objective was creating metadata as meaningful and complete as possible, with a reasonable effort and to facilitate as much as possible data searches on this metadata. Other point to be taken into consideration was that CMA users were not used to managing or creating metadata, so they needed clear guidelines, education and a simple approach to metadata (it is expected than as they mature as metadata users, it will be easier for them to understand, complete and improve the already created metadata). Thus, the main decisions taken for this project were:

- Trying to achieve at least the obligatory ISO 19115 metadata fields (more or less 40 from the around 500 fields of the complete standard) when possible.

- Using Dublin Core standard as a guideline for minimum metadata.

- Giving CMA users seminars, and spreading adequate tools, in order to promote metadata usage. The chosen tool for metadata creation was a software developed in our research lab called CatMDEdit, that allows for creating metadata and exporting/importing to/from the different relevant standards (ISO, FGDC, Dublin Core), and working against XML files, or databases (MS. Access, Oracle) [181].

Implementation

This section describes the design of the CMA SDI from the point of view of the software components deployed and will give an overview of its implementation. Some design decisions are commented and other interesting issues arisen during the development of this project are stated.

The deployment diagram of the CMA SDI is shown in figure 3.3. This diagram shows the different components used and their interactions. As the figure shows, two Compaq ProLiant servers are currently dedicated to this infrastructure. One of them, carballo.xunta.es, has the database and closely related elements (data, metadata, and

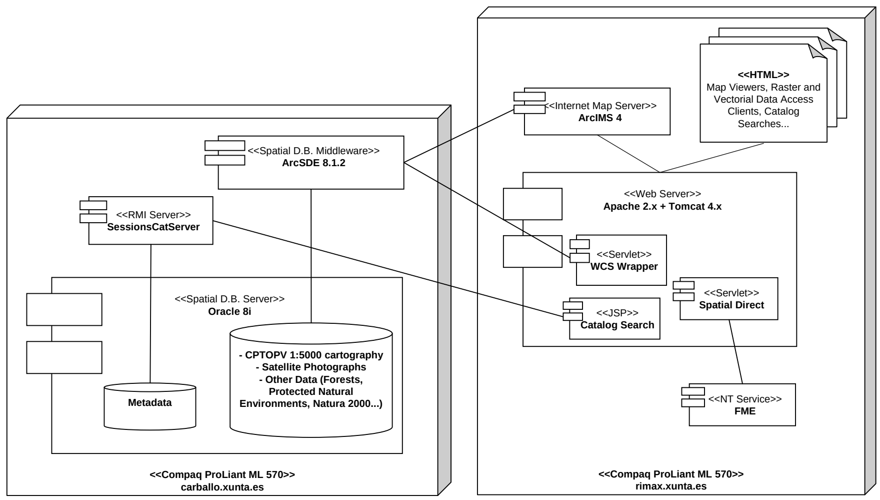

Figure 3.3: Galicia CMA SDI deployment diagram

ArcSDE). The other one, rimax.xunta.es, contains the web server, the Internet map server (ArcIMS) and the software for downloading vectorial data (Spatial Direct). These servers are currently behind a firewall, and thus are only visible in the CMA Intranet.

Applications

As the metadata catalog can be seen as the central element of an SDI architecture, this will be the starting point to describe the applications. In order to provide data searches in this catalog, a thematic search engine was developed. This search engine provides an interface that combines themes, areas, scales and dates to allow for customized data searches (figure 3.4). An example list of the results produced by a search can be seen in figure 3.5. This list shows some metadata for each dataset found (title, abstract,

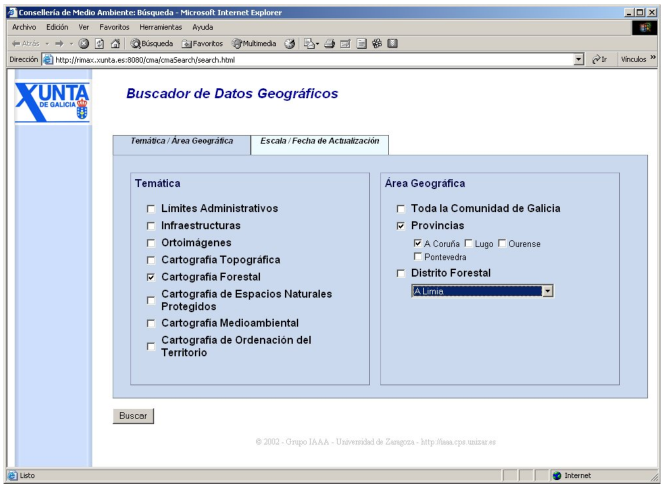

Figure 3.4: Search engine interface

scale, format, date and producer).

Clicking the title of an item opens a new window with the complete metadata for that element as shown in figure 3.6. The main advantage of the SDI architecture is the use of combinable services. This was proved true when users asked for a way to allow for a connection between data searches and map services, allowing thus to make a search in the catalog and, once found an appropriate dataset, directly access to the map services that showed it. This was implemented by providing a link for each item in the search results list that opened a window where the available map services for that item where shown (see figure 3.7). An extension of the catalog was included to support this functionality, but aiming at the implantation of a services catalog in a near future to support this and other functionalities.

After finding an adequate map service for their needs, users may access to a map viewer showing this service by following a link. An example is shown in figure 3.8, where a map service showing Galicia provinces and highways is shown. This is the ArcIMS HTML map viewer, with a little customization to fulfill CMA users' needs. Users may stop here, if geodata visualization is their only need, or they can download the data being shown. A link is provided in the map viewers to access Spatial Direct download

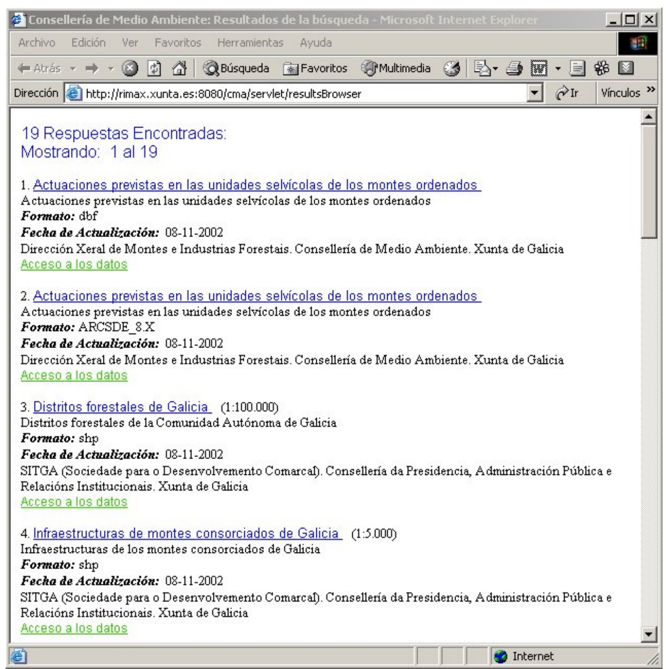

Figure 3.5: Search results

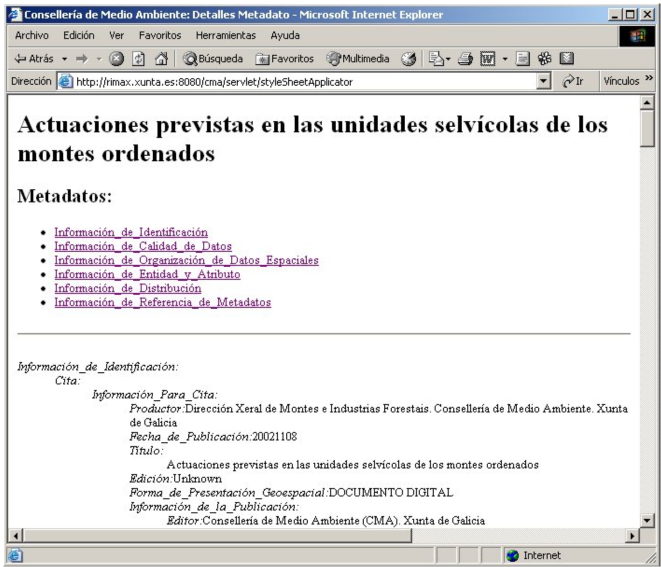

Figure 3.6: Metadata returned by search

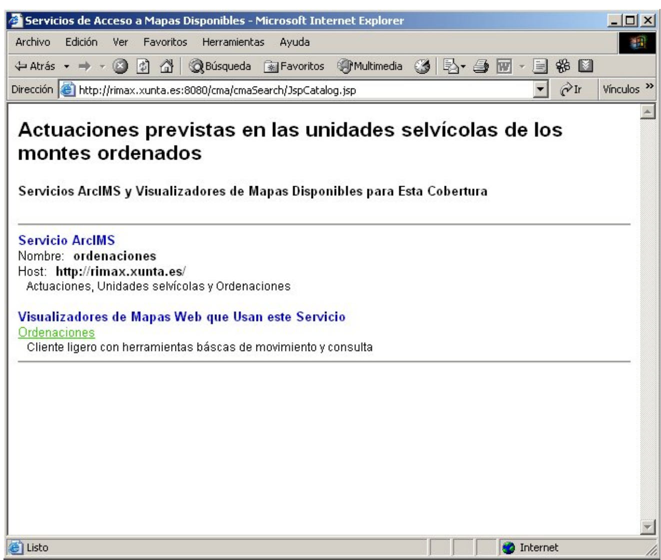

Figure 3.7: Available services

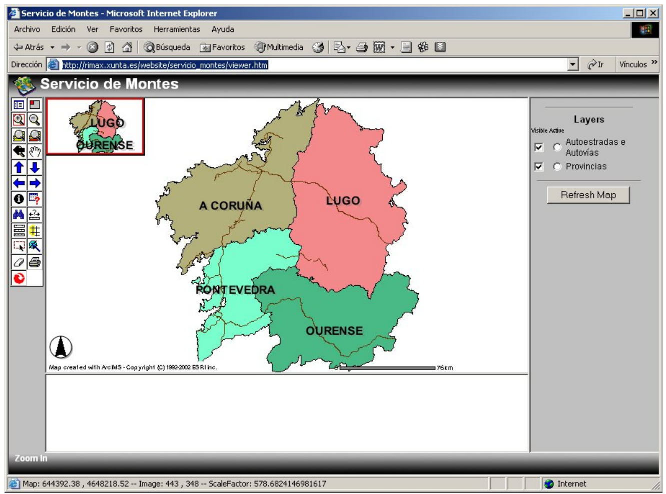

Figure 3.8: Map viewer

form (figure 3.9), already customized to provide the area and the data currently selected in the map viewer. This way all services (search, view and access) are connected (chained), giving the users an integrated view of all the SDI elements.

Final Remarks

The Spatial Data Infrastructure developed for the Department of the Environment of Galicia (CMA) has been designed and implemented to address the typical geographic information management and use problems found in big, decentralized companies and public administrations, in particular those found in the CMA, while following INSPIRE recommendations. Forest management activities have provided user requirements [19].Following INSPIRE recommendations on architecture and standards has proven to be an adequate strategy, both from a technical and a strategic perspective:

- The combinable Web services architecture has allowed for an easier integration of all elements in the infrastructure, while preparing the system for its future integration in bigger initiatives.
- The emphasis in putting a metadata catalog in the heart of the infrastructure,

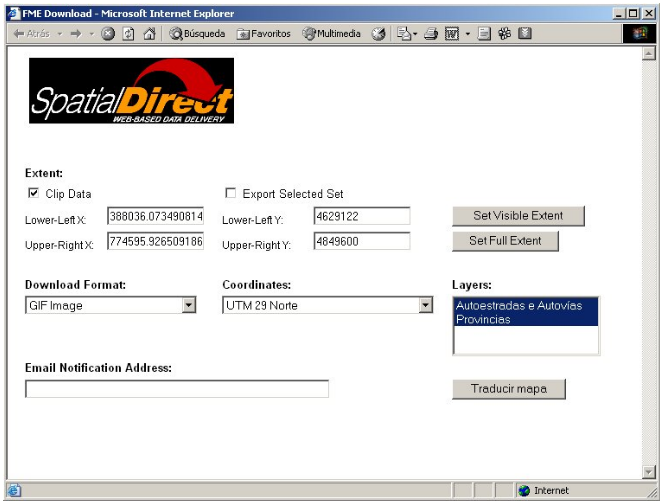

Figure 3.9: Spatial data download form

has shown is usefulness both allowing for a richer semantic description of data, thus encouraging its proper use, and giving a central component to organize the others around.

- CMA's is the first SDI initiative in Galicia. Having chosen INSPIRE recommendations as guidelines, it is now in an unbeatable strategic position to become the core of an SDI for that region, when these recommendations evolve into Community legislation.

- The COTS based approach has made it possible to develop the Galicia Region SDI, form scratch, in a six-month period. The COTS approach to the development of the SDI also had allowed the developers to take full advantage of all the previous knowledge they had about the products they were integrating.

Although currently for internal use, this SDI has been designed to allow for its immediate opening when needed. All implemented map and access services can be accessed through standard OGC Web Map Server and Web Feature Server interfaces. The catalog also follows OGC standards so it will be able to interoperate with others when needed. The appropriate technical steps have thus been given to make this SDI the core of a future Galician one, compatible with INSPIRE legislation.

Analysis Under the C&C Architectural Style

The architectural view of the Galicia CMA SDI, referred to as a 'Service Oriented Architecture', is shown in figure 3.2.

This architecture is depicted in a layered way, focusing on its components and some of their properties. It must be noted that there are neither explicit connectors nor constraints in this diagram, but some of them are detailed in the text of the previous sections. Regarding the components, and following their function as explained there, they all can be matched to some of the component types proposed in the SDI style:

- In the layer 'Data and Metadata Sources':
  - Vector Data and Raster Data are **Spatial Dataset Repositories**.
  - SDI Documentation is a **Shared Data Repository**.
  - Metadata is a **Metadata Repository**.
- In the layer 'Chainable Services':
  - WMS-Core, WMS-Raster Core and WMS-Environmental are **Portrayal Services**.
  - WFS-Core, WFS-Environmental and WCS-Raster are **Access Services**.
  - OGC Metadata Catalog and OGC Services Catalog are **Catalog Services**.
- In the layer 'Integration Services':
  - Access Control is a specialized **Application Service**.
- In the layer 'User Applications':
  - All components in this layer are **Applications**.

Some connectors and constraints can be extracted. At least there is one that is quite clear and that also matches the proposed style:

- ‘User applications are built *on top of distributed services*’. This implies a connector between user applications and distributed services and follows two of the defined constraints:
  - Clients **Request** from Servers. Applications in the SDI style are Clients and the SDI Services are Servers, so Applications Request from the SDI Services, as it happens in the CMA SDI architecture.
  - Data Accessor **Reads and Writes Data** from Shared-Data Repository. These constraints imply that any component type that is not a Data Accessor can not read or write data from Shared-Data Repositories. Neither Applications in the SDI style nor ‘User Applications’ in the CMA SDI architecture are Data Accessors. The reason is that they do not access ‘Data Sources’, they access ‘Services’.

Finally, it is worth noting that in this architecture every component has the properties suggested for the SDI style (**name**, **type** and **types of data stored** for repositories) but the connectors do not have any.

3.3.2 Architecture of the Piedmont local SDI

SITAD is the name of a project which points towards the creation of a local SDI in the Piedmont region, Italy. Designed according to the INSPIRE principles, it aims at facilitating the coordination of public sector organizations to collect, manage, distribute and reuse spatial data [44]. That paper describes the components in the SITAD and provides the architecture diagram shown in figure 3.10. Although it is not indicated whether this diagram follows some existing architectural view type, it is stated that it ‘represents the presentation logic, the business logic and the data logic of the infrastructure’ (p. 4). According to the architectural principles in [45] all that information should probably have been distributed among several views (i.e. in the same diagram are shown elements quite different like Web servers (software components) and metadata records (datasets)). Anyway, the information in this diagram and the text of the paper enable the evaluation of some elements in the SITAD architecture. These are the components described in the paper mapped, as far as it has been possible, to their equivalent types proposed in the SDI style:

- Application to compile metadata is an **Application**.
- Metadata catalogue (MTD in the figure) is a **Catalog Service**.
- Unique catalogue gateway is a **Geoportal**.
- Web map services are **Portrayal Services**.
- Download services are **Access Services** or **Information Management Services** if they hold non-spatial data.
- Visualisation services are **Information Management Services** if they show non-spatial data.
- Multi-map service viewer is an **Application**.
- User interfaces (i1 and i2 in figure 3.10)) are **Applications**.
- DBs (from the figure) are **Dataset Repositories**. When they have the 'Spatial Box' over them they are **Spatial Dataset Repositories**.

With regards to connectors or constraints, there is little information that can be extracted from the paper. The text mentions that data are accessed via on-line services and served to clients, what points out that there must be connectors between data and services (at least **Data Reading**) and between services and clients (**Request/Reply**). Probably this also implies several of the constraints defined for the SDI style, though trying to specify this would be pure speculation. There are also some connectors portrayed in figure 3.10, which seems to confirm this interpretation of the text.

The only property that is shown for some components is their **type**. There are not any properties for the connectors.

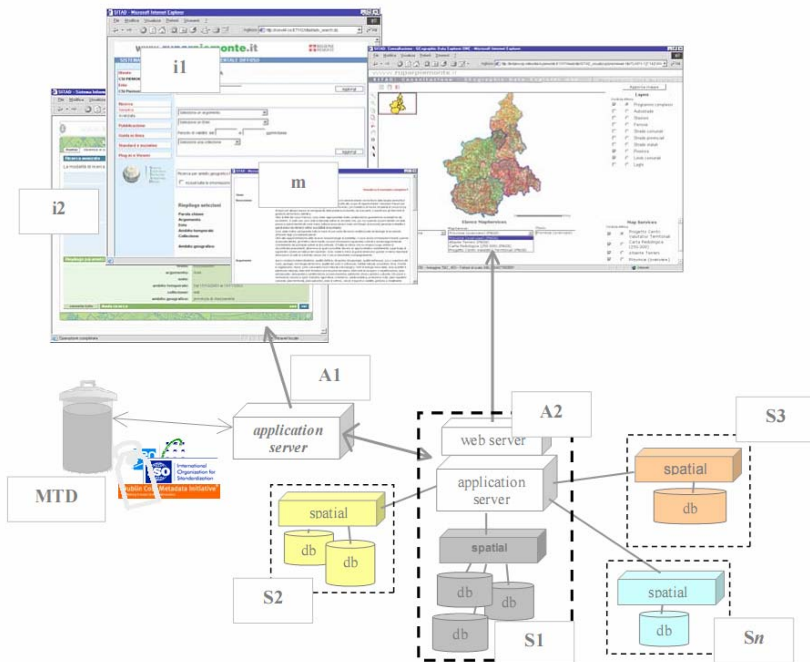

Figure 3.10: Architecture of SITAD infrastructure (taken from [44] p. 5)

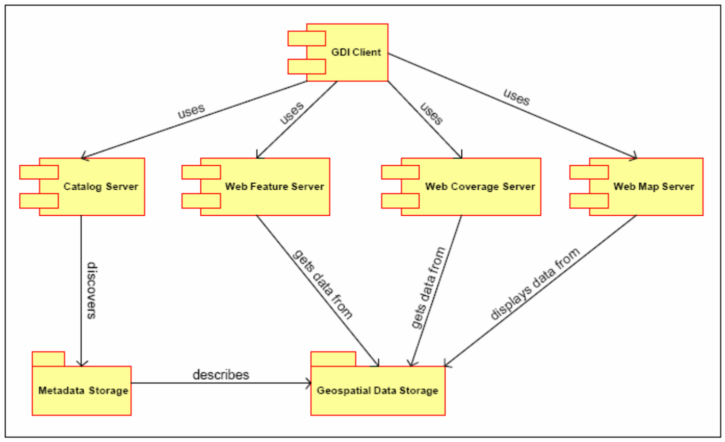

Figure 3.11: GDI Northrhine Westphalia Component Diagram (taken from [39], p. 32)

3.3.3 Architecture of the Northrine-Westphalia GDI

As described in [39], the Geospatial Data Infrastructure Northrhine-Westphalia (GDI-NRW) is an initiative of the Land Northrhine-Westphalia, in Germany. It started in January 2000, with the objective to develop a market for geographic information in that Land by connecting users, service providers and enablers, integrators, data producers and infrastructure providers. In addition to a general description of the objectives of this SDI, this paper includes an architecture model with a taxonomy of services and technical components (pp. 31-33). The component diagram is shown in figure 3.11. Although this diagram presents an architectural model and not an architectural view, it does not make it a less valid or relevant reference for the purpose of verifying the applicability of the SDI style in real projects.

First of all, instead of defining a taxonomy of services, the GDI-NRW service taxonomy adheres to the one described by ISO/TC 211 on geographic information services [92]. Then, focusing on the technical components, they define a model based on Web services, with a number of components which support them. These geospatial services are classified into three categories:

- GDI-NRW Search and Discovery Services: organization, discovery and access of geospatial information.

- GDI-NRW Access and Retrieval Services: access to geospatial information outside the scope of the catalog services.

- GDI-NRW Web Mapping Services: distributed Web mapping.

The components described in figure 3.11, following OGC specifications, fall into these categories:

- Catalog Server: search and discovery of geospatial data and services through its metadata.

- Web Map Server: services for distributed Web mapping.

- Web Coverage Server: services for access to coverage data.

- Web Feature server: service for access to feature data.

The other component types in the figure are the clients, which access any kind of data distributed in the GDI-NRW through the services, and the metadata and geospatial data storages, which are not defined though some comments are given regarding to their contents.

The paper ends giving some future steps to the architecture model, which include (with little detail) services for portrayal and presentation, ordering and payment, security, authentication, gazetteers and an e-commerce framework.

Regarding the component types in figure 3.11, and understanding their function as explained in the paper, they all can match the component types of the SDI style:

- GDI Client is an **SDI Client**.

- Catalog server is a **Catalog Service**.

- Web Feature Server and Web Coverage Server are **Access Services**.
- Web Map Server is a **Portrayal Service**.
- Metadata Storage is a **Metadata Repository**.
- Geospatial Data Storage is a **Spatial Dataset Repository**.

The proposed services for the evolution of the architecture model can match those in the SDI style, though there are some aspects that need to be clarified:

- Portrayal and Presentation Services are **Portrayal Services**.
- Gazetteer is a **Gazetteer Service**.
- Ordering and Payment Services are **SDI Services**. These kinds of services are quite specific and important and it could be argued that they should have been included in the style. The problem is that we are far from a consensus on the e-commerce technical aspects of an SDI. Although this issue is important, and addressed in some high level SDI specifications and regulations (i.e. in the INSPIRE directive text), the idea behind the proposed style is to capture, refine and systematize the existing knowledge about SDI architecture. The e-commerce issue has not been defined or implemented to an extent that makes this viable. On the other hand, the SDI style does not prevent an SDI architecture from having e-commerce services, which would extend the SDI Service type, and maybe others (i.e. **Access Services**).
- Security and Authentication Services are **SDI Services**. With these kinds of services arises a problem that is very similar to the one discussed in the previous point.

From the text of the paper and figure 3.11, some connector types and constraints can be extracted for the GDI-NRW. There are five kinds of relationships in the figure:

- GDI client *uses* Catalog Server, Web Feature Server, Web Coverage Server and Web Map Server: this one is called a **Request/Reply** connector in the SDI style.

- Web Feature Server and Web Coverage Server *get data from* Geospatial Data Storage: this one would be equivalent to the **Reads Data** connector.

- Web Map Server *displays data from* Geospatial Data Storage: this is also equivalent to **Reads Data**. The paper does not give any indication of the difference between this connector and the *get data from* discussed before.

- Catalog Server *discovers* Metadata Storage: there are no explanations about this connector, but most probably it does not mean that the Catalog Server needs to discover the location of the metadata it serves! Indeed, it seems that this connector is similar, if not identical, to the *gets data from* in the diagram, so equivalent to **Reads Data**.

There is another connector in the figure, which helps to illustrate the problems of creating an architectural diagram without defining its view type: the Metadata Storage *describes* Geospatial Data Storage connector. If the diagram is a style of a viewtype similar to the C&C, what seems implied in the paper, then the connectors should be among components, not among other elements. Although the depicted type of connector is undoubtedly present (i.e. some metadata in the Metadata Storage will surely describe some data in the Geospatial Data Storage), it is clearly different from the others, because it does not show a connection between components: probably it would be better placed on another diagram, with a different view type.

With regards to constraints, the diagram shows a layered architecture with connectors that seem to enforce some of the constraints defined for the SDI style: clients in the GDI-NRW component diagram only use servers (Client **Requests** from Server), and only the servers are allowed to get data or metadata from the storages. Therefore, it could be assumed that because servers in this diagram are all **Data Accessors** and storages are all **Shared Data Repositories**, the constraint Data Accessor **Reads Data** from **Shared-Data Repository** is implicit.

The only property that is shown, both for the components and the connectors, is their **type**. Since this proposal is an architectural model instead of an actual architecture, and so it includes component types instead of components, this is the only property that makes clear sense.

3.4 Application of the style to the Galicia CMA SDI

This chapter would not be complete without an example of application of the proposed style to document a view of an SDI architecture. Thoroughly documenting the views of a software architecture is a complex task (see [45] pp. 317-322 for some guidelines) far from the intention of this chapter; this section is focused on the *primary presentation*, as defined in that book. There are many different options to document views, from formal architecture description languages (ADLs) to various graphical notations. UML has been chosen because it is widely extended in the information systems community in general, and in the geospatial and SDI community in particular. As UML can be used in different ways to document an architecture view, some clarification is needed: objects will represent the different components in the view and associations among them will represent the connectors; different shapes have been used for the different types of objects (UML graphical stereotypes). Topological constraints are implied in the diagram (i.e. component types that must not be connected, will not be connected).

The Galicia CMA SDI has been chosen as the example to avoid defining a new project environment. Since this architecture has been found to extend some of the component types in the SDI style, figure 3.12 has been included to facilitate the understanding of the architectural view that comes next. In that figure, classes represent component types, and those on top are the component types defined for the SDI style. In the rest of this section this question will thus be answered: if the SDI style had been followed, how would a view of the Galicia CMA SDI architecture have been documented?

Figure 3.13 shows an architecture view of the Galicia CMA SDI, following the guidelines given by the SDI style. Several components projected, but not implemented, have not been included in order to have a diagram easier to understand. All the elements shown have their type: components have names and are of a type defined in the SDI style, or of a type which extends one in the SDI style. Repositories include the data types they hold (one of their properties). Connectors are explicit: they have

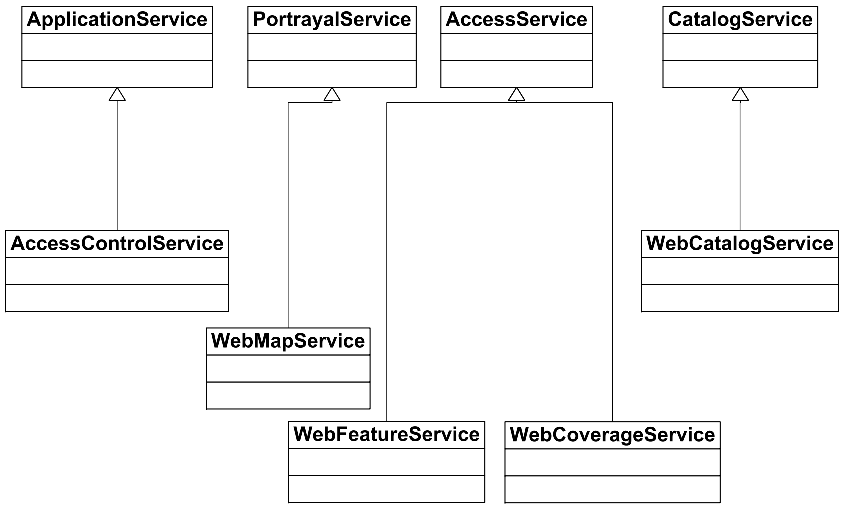

Figure 3.12: Galicia CMA SDI component types which extend those in the SDI style

a name and their shapes indicate their type. The question is: what are the differences between this diagram and the one shown in figure 3.2?

- Once the SDI style is known, the meaning of this diagram is better defined. Most component and connector types, except for those defined specifically for the project, have a defined meaning. Even from those that are not defined in the SDI style, i.e. the Web Map Service, things can be immediately deduced: for example, as the Web Map Service extends the **Portrayal Service**, everything that is true for the **Portrayal Service** (definition, constraints etc.) must also be true for the Web Map Service.

- This diagram is more complete: connectors are explicit, and also the types of the components. For example it is now clear that the services do not write to the repositories, only read from them.

- As constraints are explicit for the SDI style, one can be sure that they are fulfilled:

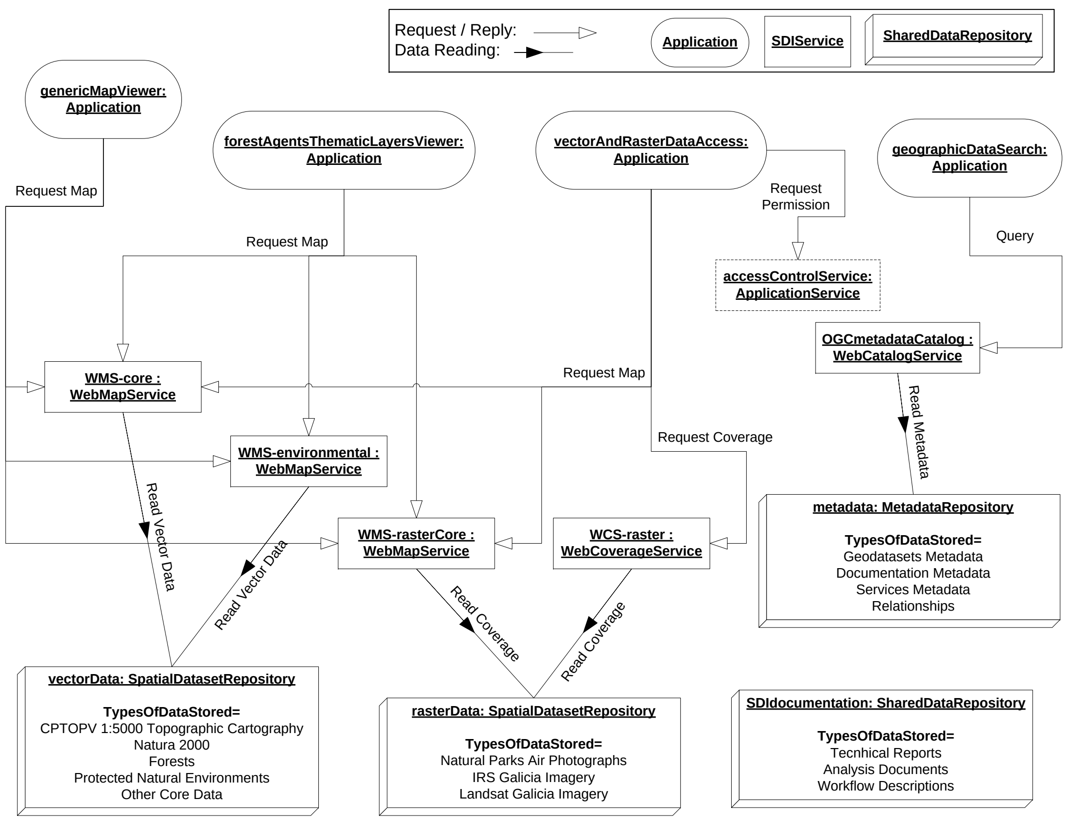

Figure 3.13: Galicia Architecture View following the SDI style

for example, it is clear that the applications in the Galicia CMA SDI do not read data from the data repositories (this was not so clear before).

3.5 Conclusions

This chapter proposes a pattern to design and document distributed geographic information systems following SDI design principles. The pattern has been presented as an architectural style, defined under the component-and-connector view type, extending two well-known styles in distributed information systems: the client-server and shared-data styles. The style has been created analyzing several important SDI architecture proposals, finding their common elements, and giving them a unique name and a definition. Several elements that a software architecture should consider, which had not been properly addressed in these proposals yet, have also been discussed and included in the style (specially connectors and constraints). Three real SDI projects, with published architectural views or models, have been examined to verify whether the style would have been applicable to them. For one of these projects the style has been effectively applied to show how this could have been done.

The proposed style offers a systematization, refinement and extension of knowledge about SDI architectures, and it is grounded on well-known concepts in software architecture. This can help the designers of an SDI to start their design with some clear guidelines, to exchange knowledge about SDI architectures, and to clarify what an SDI and its architecture are. The style has been defined with its extension in mind: it is a minimum core of elements that are common to most SDI proposals, either explicitly or implicitly, but a system architect may extend it to address specific necessities of an SDI. Indeed, there are several aspects of SDIs that the style does not address: e-commerce, security etc. These are issues which are currently under discussion, so it was considered that it was too early to include them.

There is another issue that must be taken into consideration. There is a refinement of the SDI style that could have been considered: the use of OGC and ISO specifications for the components in the style when possible (i.e. instead of a **Portrayal Service**, a Web Map Service could have been included). Although this was seriously considered, a more abstract approach was decided. This decision was adopted to promote concepts before technologies and because the value of a style is larger when it can be applied to more architectures; specifying too much detail reduces its applicability. The result is a style that can be easily refined to allow only for OGC components, but which does not force them. In addition to this, neither ISO nor OGC have defined every component type in the SDI style, and almost none of the other elements (connectors, properties and constraints), so the style could not have been completely defined in OGC or ISO terms.

To finish these conclusions, it is important to remark two issues about the scope of this work. First of all the style proposed, included in the component-and-connector view type, gives only one view type for SDIs. There are other view types and styles for software systems that would be interesting for SDIs; for example the deployment style, in the allocation view type, which can be used to analyze certain properties of a software system (i.e. performance, as explained in [45]). The second issue is that as well as being distributed geographic information systems, SDIs are also Information Infrastructures, composed by different independent systems working together. From this point of view their architecture can not be designed, but 'cultivated' [72]. This will require further advances that allow us to analyze their architectural properties in terms of the systems that compose them, and not only in terms of their components.

Chapter 4

Contributions to the Modelling of Spatial Data Infrastructure Portrayal Services

4.1 Introduction

Portrayal services, usually following the OGC Web Map Service (WMS) specification, are among the main software components of any SDI. They are designed to facilitate users to browse maps on the Web without the need of specialized applications. The purpose of this chapter is to discuss some aspects of the design of a Web Map Service implementation which has been in development and use since 1999. This implementation, called JMapServer, is built on top of JGISView, a Java GIS visualization component and desktop application.

After introducing some aspects related to GIS visualization, Web mapping and the OGC WMS specification, the core elements of JGISView design are presented. The next section describes an analysis pattern to implement map labelings, where labels include anything that can be portrayed on a map and created with the information this map shows. Then the architecture and certain relevant design aspects of JMapServer are described. Before the conclusions of the chapter, two examples of applications developed with JGISView and JMapServer are shown in order to validate its applicability to real problems and their role in development environments where SDI services are available.

4.1.1 Maps and Geographic Information Systems

According to [150] maps have been the central focus of the science of cartography for many years. They provided a storage for needed information and also a picture of the real world. But computers came to change that: digital cartography has replaced the printed map with digital databases as the storage medium for geographic information and cartographic visualization is the means to picture this information as a map.

A map is the graphic representation of a geographical setting. The main task of maps is to communicate spatial relationships and forms. Maps are concerned with two elements of reality: locations and attributes. Locations are positions, normally coordinates in a two-dimensional space, and attributes are qualities or magnitudes, such as languages or temperatures. Maps are reductions and abstractions of reality: they are smaller than the region they portray, by a scale factor, and portray only those elements of reality that have been chosen to fit the use of the map. Maps use symbols to stand for elements of reality. These symbols consist of different kinds of lines, dots, colors, tones, patterns etc.

There are several categories of maps: according to the scale (the larger the scale the smaller the portrayed areas with a better detail), or the function (general reference maps, i.e. topographic maps, thematic maps, which concentrate on the distribution of a single attribute or the relationship among several, i.e. temperatures or population, charts, used for navigation) or the subject matter (cadastral maps with land holdings and their owners, or plans with buildings, roadways and boundary lines).

Longley et al. [114] introduce the concept of representation as a model of some part of the real world, built by humans to assemble knowledge about our planet and to serve for planning, resource management, conservation, travel etc. This concept can be moved to the computing world: digital representations of geography hold clear advantages over previous types like paper maps: we can use the same digital devices to handle every type of information, digital data are easy to copy, fast to transmit, can be stored in small spaces and are less subject to physical deterioration. More importantly, digital data are easy to transform, process and analyze.

The two fundamental ways of representing geography are discrete objects and fields.

Discrete objects have well-defined boundaries and are instances of generally recognized categories or classes. In the field view, the world can be described by a finite number of variables, defined at every possible position and which change in value across the Earth's surface. The digital representation of these conceptual views is provided by vector and raster data respectively.

Vector data are formed by points. These points, lines, which are sequences of connected points, and areas, series of vertices connected by straight lines, are the basic geometrical representations of discrete objects as vectors. Raster data are typically arrays of cells, usually square. These cells have attributes, or properties, that express geographic variation. If these properties are colours, cells are typically called pixels. Both kinds of digital representations are usually georeferenced, located somewhere on Earth, typically by means of coordinates.

Spatial objects are usually classified as *points*, for spatial occurrences or events, *lines*, represent linear entities as roads, *areas*, two dimensional entities that represent natural (agricultural fields) or artificial aggregations (parcels) and *surfaces or volumes*, for three dimensional natural objects, as river basins, or artificial phenomena, such as the population potential of metro stations. These spatial objects are usually linked to their attributes, which are alphanumeric data describing them.

Longley et al. [114] state that paper maps have several limitations: fixed scale, fixed extent, limited to a static, two-dimensional and usually complete, i.e. not supplementable, and map producer-centric view of the world. All these problems do not arise in digital maps produced in computers by means of geographic information systems (GIS). Anyway, the use of GIS does not change the requirements of good mapping, which requires that spatial objects and their attributes can be readily interpreted in a visual way. The use of graphic symbols allows to communicate different types of information associated to the attributes of spatial objects. Several properties of these graphic symbols can be employe (size, value, hue, saturation, orientation, shape, arrangement, texture, focus) to achieve this. A related task is how best position graphic symbols on the map, to optimize its interpretability. Good placement of symbols requires the avoidance of overlap, and proper alignment and positioning.

4.1.2 Web Mapping

Peng and Tsou [140] define Internet GIS 'as network-based geographic information services that utilize both wired and wireless Internet to access geographic information, spatial analytical tools, and GIS Web services' (p xxx). As described in section 4.1, an important application of GIS is mapping, so it can be easily deduced that an important application of Internet GIS will be Internet, or Web, mapping. Indeed Internet GIS started when users first published graphical images of static maps in Web pages, evolved to true Web mapping, with increasing interactivity as the technology advanced, and has achieved the current state with the availability of many different types of distributed, many of them standardized, GIS services. With a focus on Web mapping, there have been thus three generations:

- A first generation, static web mapping, started with what is probably the first Web map viewer, the Xerox PARC Map Viewer [145], developed in June 1993. In this generation, all map processing occurred in the Web server and the viewer, or client, supported only basic map interaction (zooming, selecting data to be shown, etc.). Clients were simple HTML pages that required only basic Web browser capabilities.

- A second generation, interactive web mapping, appeared as Web technology evolved: through the use of Dynamic HTML (DHTML), client-side scripting, the Document Object Model (DOM) exposed by Web browsers and Cascading Style Sheets (CSS) dynamic and interactive applications were developed on Web browsers. Plug-ins (Java applets, Microsoft ActiveX controls etc.) allowed even higher levels of interactivity.

- The third generation most prominent characteristic is standardization. The Open Geospatial Consortium (OGC, http://www.opengeospatial.org) Web mapping activities started in 1997 and eventually led to the development of the Web Map Service (WMS) interface specification, which is also approved by the ISO Technical Committee 211 (ISO TC 211) as an international standard (ISO 19128). The WMS specifies protocols that provide uniform access to those Internet map servers which support them. Any client software complying to the specification allows automatic overlay of map images from several of these map servers in spite of their internal implementation details. This specification is explained in more detail in section 4.1.3.

It is not clear if the existence of services and APIs like those provided by Google Maps (http://maps.google.com), Yahoo! Maps (http://http://maps.yahoo.com/) or Microsoft Live Maps (http://maps.live.com/) sets up a fourth generation. Technically these types of services are similar to those in the third generation, though they use pre-rendered maps divided in tiles to improve efficiency [149]. Maybe the fundamental difference is in the data. As suggested by Tim O'Reilly in his well-known essay about the 'Web 2.0' [136], the main differences are the fact that these online services allow to access updated maps of a significant part of the World with a very good level of detail, and the fact that they are designed to allow users to easily employ them in new applications, i.e. 'mash-ups'.

4.1.3 The Web Map Service Interface Specification

The WMS Interface Specification [100] establishes an interface for Internet map servers. For the sake of simplicity, and when the meaning is clear, in the rest of this chapter the acronym WMS will be used to refer to this specification, and also to any Internet map server that complies with it.

The WMS specification defines a map as a digital image which portrays geographic information. It is important to notice that the returned map is just an image: this implies that the geographic information used to create that image is never sent to the client that requests the maps, only images, in typical image formats like PNG, JPEG etc., of this information are sent. There are three operations in the WMS interface, the third one optional:

- **getCapabilities**: it provides service-level metadata about the WMS.
- **getMap**: it returns a map, a picture, as requested. The client of the service can indicate which information must be included in the map, the geographic

extent, the size of the returned image etc. The geographic information available to portray maps with a WMS is classified in *layers*, which have *styles* that specify how to render them.

- **getFeatureInfo**: it returns alphanumeric information about features shown on a map.

WMS operations are defined as simple GET or POST HTTP requests. Another important feature is that when several maps are produced with the same geographic parameters, and with an image format that supports transparency like the PNG format, they can be overlaid to produce a composite map. Finally is necessary to point out that the WMS specification does not support user-defined symbolization of maps, i.e. their style is predefined, but there is a profile of the WMS specification [117] that allows to do this by allowing to use the Styled Layer Descriptor (SLD) [107], an XML based language to describe cartographic symbols.

4.1.4 SDI Portrayal Services

WMS play an important role in SDIs, because they are the usual recommendation to set up their portrayal services. Online mapping, and the WMS specification, take a whole chapter in the SDI Cookbook [78, chap. 5]. All SDI proposals we are aware of include online mapping services, and most of them recommend using the WMS specification (see section 3.2.1). The architectural style proposed in section 3.1 includes the Portrayal Service as an abstraction of the WMS type.

There are more than five hundred implemented instances of this service listed in http://wms-sites.com, what can be seen as an example of its widespread use. There are more than one hundred implemented instances listed in service directory of the the Spain SDI (IDEE) (http://www.idee.es/) where, for instance, there are only four instances of an OGC compliant Catalog Service, or less than fifteen of an OGC Web Feature Service (a service used for the access to spatial data in SDIs).

4.2 JGISView: A Java GIS Visualization Component

JGISView is the name of a library of Java [160] components designed to portray geographic information, and also the name of a desktop application that allows for the interactive use of these components. The need to develop an application to view a large amount of satellite orthoimages in an Intranet environment [5] was the beginning of JGISView. This library has been developed with these objectives:

- Developing a set of components that were able to support generic GIS visualization in order to use them in different applications.
- Developing a set of tools for the use of these classes in an interactive Graphical User Interface (GUI) (see figure 4.1).
- Designing the components so it is easy to extend them to fulfil new requirements and it is possible to use them in different kinds of applications, from desktop to Internet servers.

The structure of the fundamental Java packages in JGISView is shown in figure 4.2. The main packages in this figure are:

- **Event**: Classes to manage the communication between classes, which is designed to follow the Observer design pattern [69].
- **Layer**: Contains classes to load geodata as layers, and the classes that allow the symbolization of these data.
- **Graphics**: Classes to manage geometries for vectorial data.
- **GUI**: Graphical User Interface of JGISView, designed according to the Model-View-Controller design pattern [40].
- **Map**: Classes that define maps and coordinate reference systems.

The core elements of JGISView are those in the packages Map and Layer. The structure of the most important of them and their relationships is given in the UML class diagram in Figure 4.3. These classes and interfaces are:

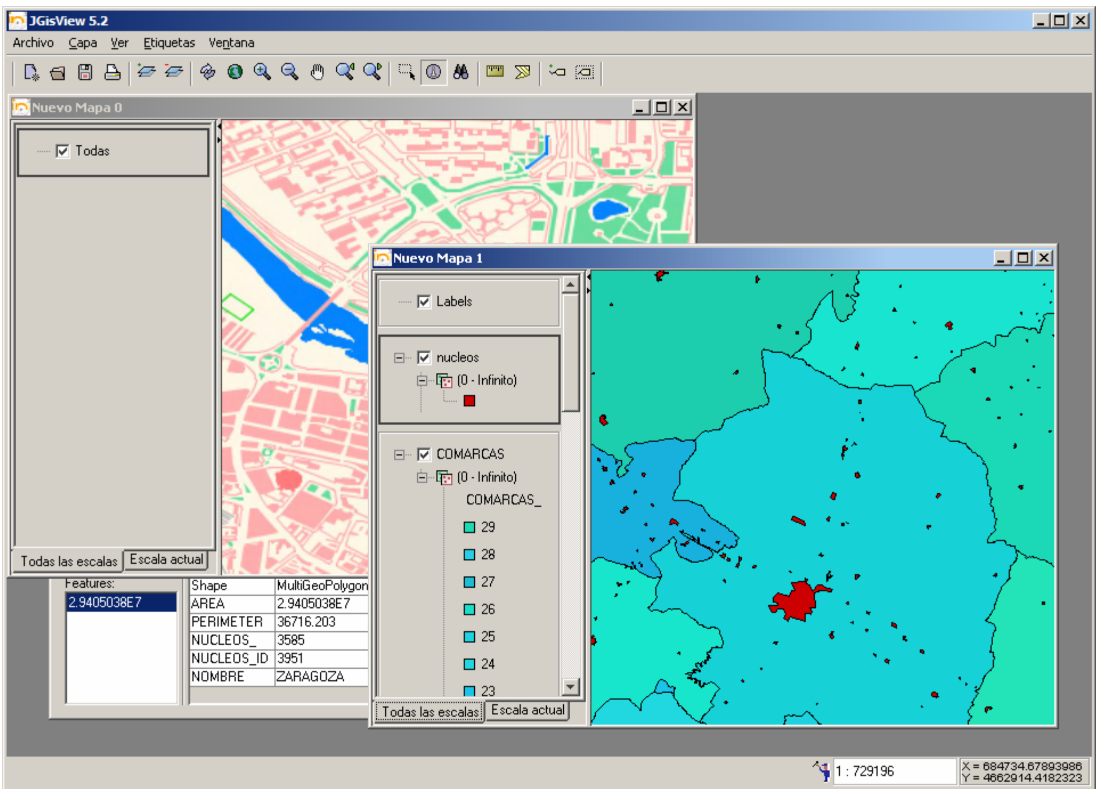

Figure 4.1: JGISView GUI

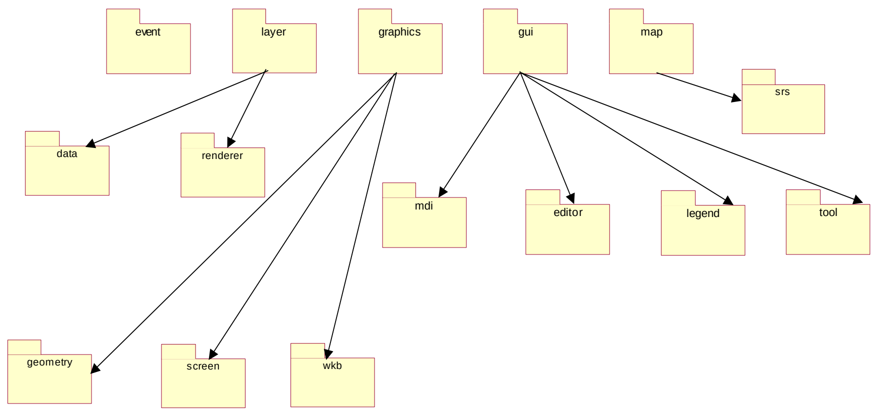

Figure 4.2: JGISView Packages

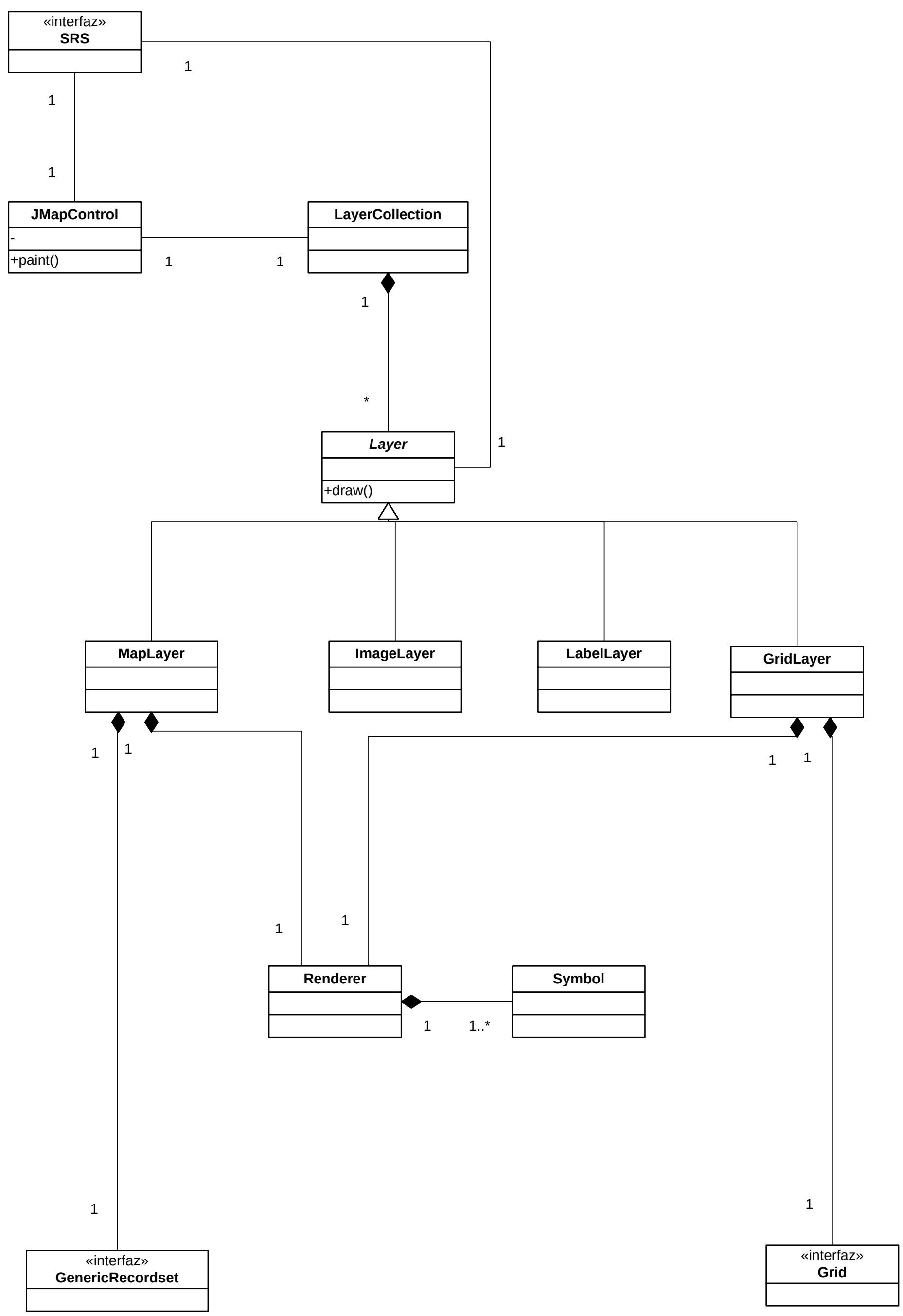

Figure 4.3: JGISView Main Classes

- **JMapControl**: It represents a map, understood as a graphic panel where a set of layers that contain geographic data are portrayed. It also holds the necessary operations to properly position geometries and images in the map, given its spatial reference system.
- **SRS**: Its subclasses define coordinate reference systems and support transformations of coordinates among them.
- **LayerCollection**: It holds the Layers in a JMapControl and manages their order.
- **Layer**: It is an abstract class which defines the common behavior of all layers in JGISView. Its main responsibility is drawing the geographic data it holds on a JMapControl.
- **MapLayer**: A layer that holds vector data (in its GenericRecordset) and the rules to portray them (class Renderer).
- **GenericRecordset**: It represents a table structure able to hold any kind of data. If one of the columns of this table is a Geometry, its contents will be able to be portrayed by a MapLayer.
- **Renderer**: A definition of rules to portray the data hold by a MapLayer or GridLayer by means of Symbols.
- **Symbol**: A class that takes a geometry (point, line or rectangle) and it is able to render it with a specified style (color, size etc.).
- **ImageLayer**: A Layer that holds a georeferenced image and is able to draw it on a JMapControl.
- **LabelLayer**: A Layer that contains all the Labels (textual or not) defined in a JMapControl and which draws them on top of it.
- **GridLayer**: A Layer that holds raster data (in its Grid) and the rules to portray them (class GridRenderer).
- **Grid**: A set of raster data defined as a georeferenced set of regular cells.

- **GridRenderer**: A definition of rules to portray the data hold by a GridLayer by means of Symbols.

To facilitate the development of applications that integrate JGISView, the use of JavaBeans [158] was decided. JavaBeans are Java software components that can be easily manipulated from the graphic designer of any development environment for Java. Several of the main JGISView classes, JMapControl, MapLayer and ImageLayer, where modified to make them Java Beans. Besides these main classes, several GUI classes, MapLegend, ScaleBox and CoordinatesPanel were also modified to make them JavaBeans. Modifications were simple, JavaBeans are normal Java classes which follow certain conventions, and the possibilities to design applications using JGISView were clearly improved. Figure 4.4 shows several of the JavaBeans inside JBuilder 3.0 in design time.

4.3 An Analysis Pattern to Support Automatic Label Placement

This section describes a contribution to the object-oriented software design techniques for the map labeling problem: the *Generic Label* analysis pattern [66]. This is presented in order to allow the homogenous treatment, in software with GIS visualization and/or printing capabilities, of text labels, statistical charts, icons and other elements that have information about the features on a map and are drawn on top of it.

It is a common cartographic technique to employ different colors and symbols in order to show thematic information on maps. For example choropleth maps, where each spatial unit is filled with a color or pattern, proportional symbols that are used with point and line spatial data, and different kinds of icons, line styles and fill patterns are all used to increase the possibilities for cartographers to show information on a map. These kind of maps are typically designed in GIS applications by creating renderers based on some attributes of the layers that compose them. When the need arises to provide more complex thematic information, i.e. based in several variables at the same time, statistical charts are often used; they provide a way to compare these spatially distributed variables by just having a look at the map. Text labels are other essential

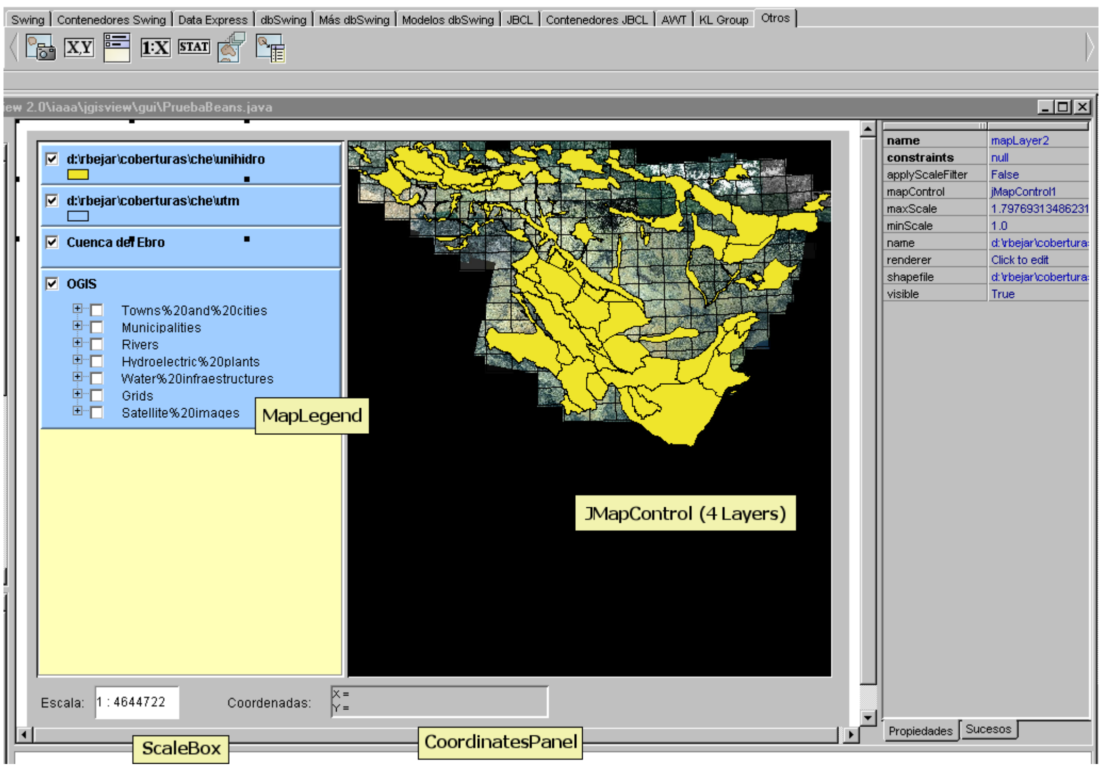

Figure 4.4: JGISView JavaBeans in JBuilder in design time

element that makes a map readable by naming the geographic features displayed on it. In fact texts are so fundamental that it is quite difficult to find maps without them. Although typically considered different elements, text labels and statistical charts have many things in common:

- They can be automatically created by the GIS package, with the visual style defined by the user and based on some feature attributes.

- They are both drawn on top of the layers that compose the map.

- There are similar requirements to efficiently position them on the map while maximizing cartographic requirements on legibility and information display.

This work proposes to manage homogeneously text, charts, icons, etc., by means of a generic labeling solution. Although many research works address the general map labeling problem (i.e. the ability to label maps with point, line and area features), most of them are purely algorithmic and do not consider its application to a variety of elements in maps [52] [34] [168] [157]. When non-textual elements are considered they are treated specially (i.e. see [170] for an algorithm that studies the placement of statistical charts in areas) or included in the labeling algorithm but treated differently than texts [82].

This work offers a software analysis pattern to design software that needs to label maps composed by point, line and polygon features, with texts and other elements (i.e. statistical charts) in a uniform way. The solution has been implemented in a Java Web Map Server and is described in detail in the next section. The algorithmic part of the map labeling solution proposed here is a simulated annealing algorithm as described in [52], though that paper applies it only to texts. This choice requires algorithms for the generation of label candidates, which are candidate places to put labels by their features, and a labeling quality evaluation procedure. For the sake of completeness this is very briefly described though the pattern presented in this work is independent from the algorithm chosen to place the labels on the map.

4.3.1 The Generic Label Pattern

Martin Fowler gives a simple and generic definition for pattern in [66] (p XV): ‘A pattern is an idea that has been useful in one practical context and will probably be useful in others’. Fowler adds more regarding to analysis patterns: ‘[...] patterns that reflect conceptual structures of business process [...]’.

This section presents an analysis pattern that has been called Generic Label. In this case, the business domain is GIS, and the pattern offers a solution to support flexible map labeling in GIS software. One important thing of patterns is that they should be discovered, not invented. The Generic Label pattern has been extracted from the second major refactoring and extension of the labeling infrastructure in a Java OGC Web Map Service in development since year 2000 [62].

- **Context**: GIS applications with visualization and/or printing capabilities.

- **Problem**: There are many examples of situations that require automatic positioning of information (i.e. labels) about features on a map:

  - Automatic positioning of text labels to create cartographic-quality maps from geodata.
  - Solutions for thematic mapping that include statistical charts, icons etc. to increase visual information on the different features on a map.
  - Moving objects over maps, that require constant identification and status information, i.e. truck fleet tracking by GPS.

  These situations often happen where interactivity and efficiency is needed, i.e. Web mapping, and in very dynamic environments, where new geodata and geoservices appear, change and disappear frequently (i.e. in an SDI, where the different actors will be providing their geodata and geoservices and will have their own updating plans and paces). In this situation, is not trivial for a GIS software to provide solutions to build fast, dynamic maps, that still show enough information and have enough cartographic quality to be useful in a wide range of uses.

- **Solution**: The Generic Label pattern provides a generic concept that allows to separate *content*, i.e. feature attributes used to create a label, *position and size*, that change dynamically, and *symbolization*, i.e. as a text string. This separation allows to represent the same label content in different ways, like the Document-View design pattern [40] does for the design of graphical interfaces, while offering support for an automatic map labeling algorithm that is independent of both label content and symbolization. This way new symbolizations and/or map labeling algorithms may be added independently without having to change the label content model.

- **Structure**: The UML class diagram that reflects the structure of the Generic Label pattern is shown in figure 4.5. The three most important classes are:

  - **GenericLabel**: An element, with information about a feature, that is positioned over a map in a certain position and with a certain size (i.e. with a certain bounding polygon). This element will show the information in a LabelContent and will be drawn by a LabelRenderer. The Map Labeling Algorithm will only need to work with these GenericLabels, thus being independent from the content or the way to draw them.
  - **LabelContent**: The information about a feature that should be drawn on a map next to this feature, whenever it is visible, and following some cartographic criteria (i.e. appropriate size and style, avoiding overlapping with other similar elements, favoring certain positions next to its associated feature etc.). Several different LabelContents may be defined for a Feature.
  - **LabelRenderer**: : It is the class that draws GenericLabels on a map. Subclassing it allows to have different symbols (i.e. texts, charts, icons) to render the LabelContents of these GenericLabels.
  - The other classes are given for context, and are simplifications chosen to show how this pattern could be used: A Map would be composed of Features, with their Attributes, and the user would associate LabelContents and LabelRenderers to every Feature. The Map would create GenericLabels

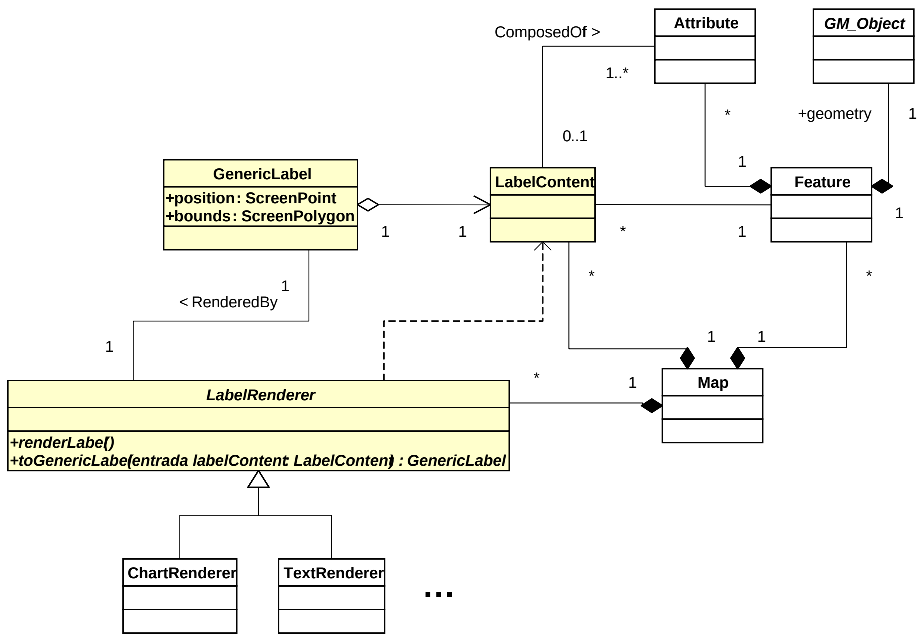

Figure 4.5: Generic Label Pattern Structure

based on these LabelContents and LabelRenderers (it needs both to calculate the size and position of the GenericLabel) and would give them to the map labeling algorithm to be positioned.

- **Dynamics**: The user defines LabelContents, with their LabelRenderers, for the Features in the Map that she wants labeled. When the Map needs to be drawn, it selects the Features in the current geographic extent. Then takes the LabelContents of these Features and gives them to their corresponding LabelRenderers to create GenericLabels (method toGenericLabel). This is so because the bounding box of a GenericLabel depends both on its content (i.e. the longer the string, the bigger the bounds) and on its renderer (a pie chart will have different size from a text string), and the LabelRenderer class has access to both content and renderer. These GenericLabels are then given to the map labeling algorithm that will move them to their final positions. The Map will finally draw these GenericLabels, by means of their associated LabelRenderers (method renderLabel, that takes the LabelContent and paints it in the position given by the GenericLabel), over the Features.

- **Uses**: An example of map labeled by a software implementing this pattern is shown in figure 4.6. In this example, Features are some USA States, and the GenericLabels are created with two different LabelContents composed one of them by three Attributes (number of white, black and hispanic people in every State) and the other by one Attribute (the name of the State). Three LabelRenderers are simultaneously used: one that shows a LabelContent as plain Text in size 10, other one that shows a LabelContent as bold Text with size 12 and finally one that renders its Content as a Pie Chart. All the GenericLabels are drawn completely inside of its corresponding State and do not overlap among them. This maps shows how the Generic Label pattern allows to automatically mix different kinds of LabelContents and LabelRenderers in the same map, while avoiding overlaps and following the cartographic criteria defined in the labeling algorithm

- **Variations**: The Generic Label pattern itself supports only one LabelRenderer

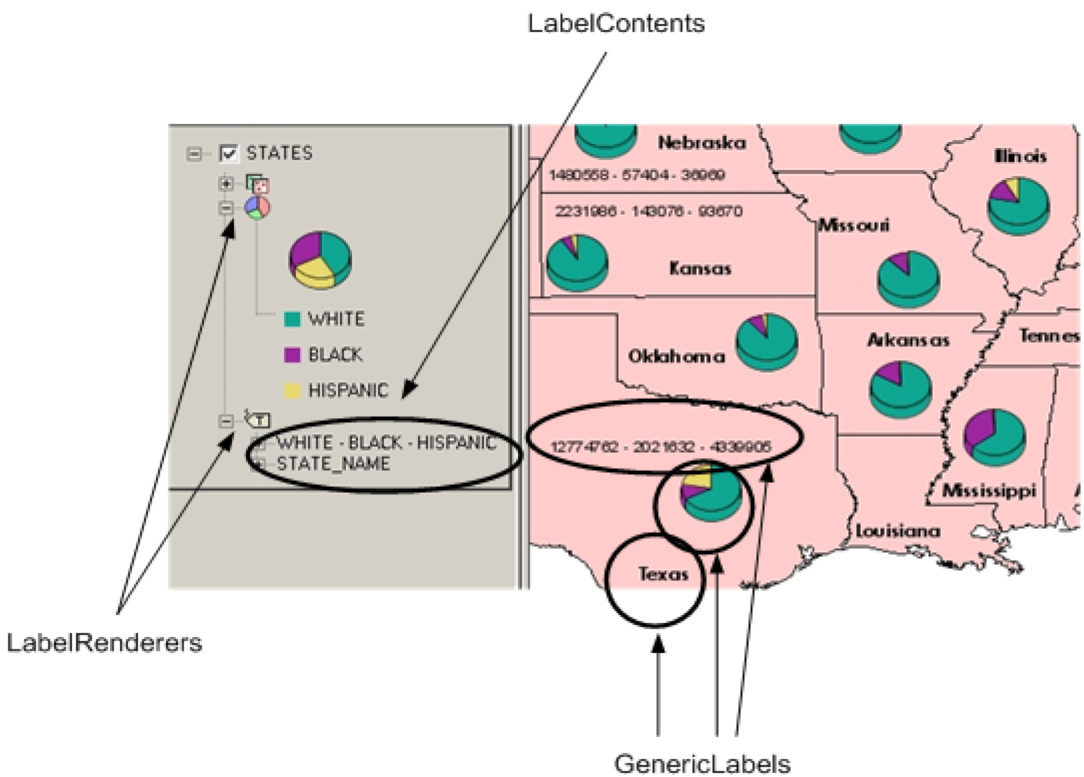

Figure 4.6: Generic Label Pattern in Action

for each GenericLabel. Several LabelRenderers can be applied to the same LabelContent by creating different GenericLabels, as shown in figure 4.6. This has the limitation that all GenericLabels will be drawn, if the cartographic quality criteria are fulfilled (i.e. if there is space for all of them), for each LabelContent. There are certain situations where this is not desirable: in example for long text labels (those composed by several words), different TextRenderers could be tried for the same map (i.e. one that draws al text in one line, and other one that partitions the text in two or more lines) to achieve the best possible labeling. In this case a label should be drawn at most once for each feature. To support this, a variation of the pattern, the MultiRenderer Generic Label is shown in figure 4.7. This variation allows a GenericLabel to have several LabelRenderers, and a different bounding area (ScreenPolygons) for each one of them. This way the positioning algorithm still receives GenericLabels (it is still independent from the content and the symbolization of a Label) but can take into consideration the fact that a GenericLabel can have different sized bounding areas, for the different LabelRenderers, and thus it can select the best one for a given map labeling.

4.3.2 Labeling a Map with Generic Labels

Generic Labels offer a solution to manage different kinds of labels on a map, but they do not provide a solution for an automatic quality labeling. A map labeling algorithm, adapted to use generic labels, is still needed. Simulated annealing, as described in [52] was chosen as the base for this algorithm. It briefly consists in calculating several candidates, possible positions to place a label, for each feature to be labeled, giving each candidate a preference level or weight, and then selecting one for each feature and evaluating the quality of the resulting labeling, repeating this latest steps until a good solution is found. Finally any remaining label overlaps are solved removing all but one of the overlapping labels.

Regarding both the evaluation of the quality of a labeling, and the generation of candidates for the labels, the solution proposed here follows some common cartographic criteria to make readable and understandable maps with labels: a good classification of

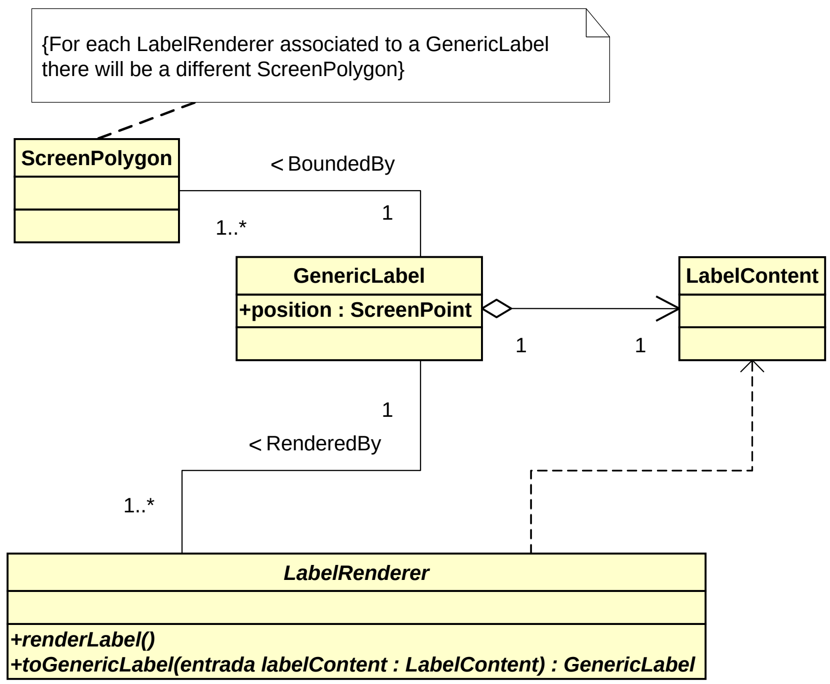

Figure 4.7: Generic Label Variation

many of those label placement rules is given in [169]. Some other more complex criteria, like trying to avoid the overlaps between labels and other features in the map have not yet been taken into consideration, as some experimental results show problems with the efficiency of this approach [190].

The idea for the evaluation function used was also taken from [52] and, briefly, is a summation of all the weights of the selected candidates for each label in a given labeling, with a penalization for labels overlapping. A variation of the penalization for labels was tried, that depended on the percentage of overlapping among them, nevertheless this idea resulted worse both in time and labeling quality.

The candidate generator algorithms define the possible places where generic labels can be positioned by their features. All these generators work properly without needing information about content or symbolization of the labels; they only need their shape and the feature they are labeling. This allows to apply them to generic labels. Although the Generic Label pattern allows any kind of polygons to define the shape of labels, rectangles have been used for efficiency reasons.

4.3.3 Implementation results

The figure 4.8 shows an example of map labeled with the proposed solution. It includes line, point and polygon labels (blue, black, in a yellow rectangle) and also pie charts, with some numeric data from the polygons (municipalities in our region). Although the cartographic quality of this map is of course improvable, we consider it as a good result for the painting times achieved. Anyway this is a first approach, and the cartographic quality achieved is a consequence of the chosen algorithm (simulated annealing): it does not depend on the Generic Label concept presented in this work, that is independent from the labeling algorithm used and thus can be adapted to get different results.

4.3.4 Generic Label Implementation in JGISView

Figure 4.9 shows a slightly simplified version of the implementation of the Generic Label pattern in JGISView. Notes attached to the classes show which JGISView classes play each role in the pattern (the pattern is shown in Figure 4.5). JMapControl, a sequence of Layers, is the central JGISView class. There is a special kind of Layer, LabelLayer;

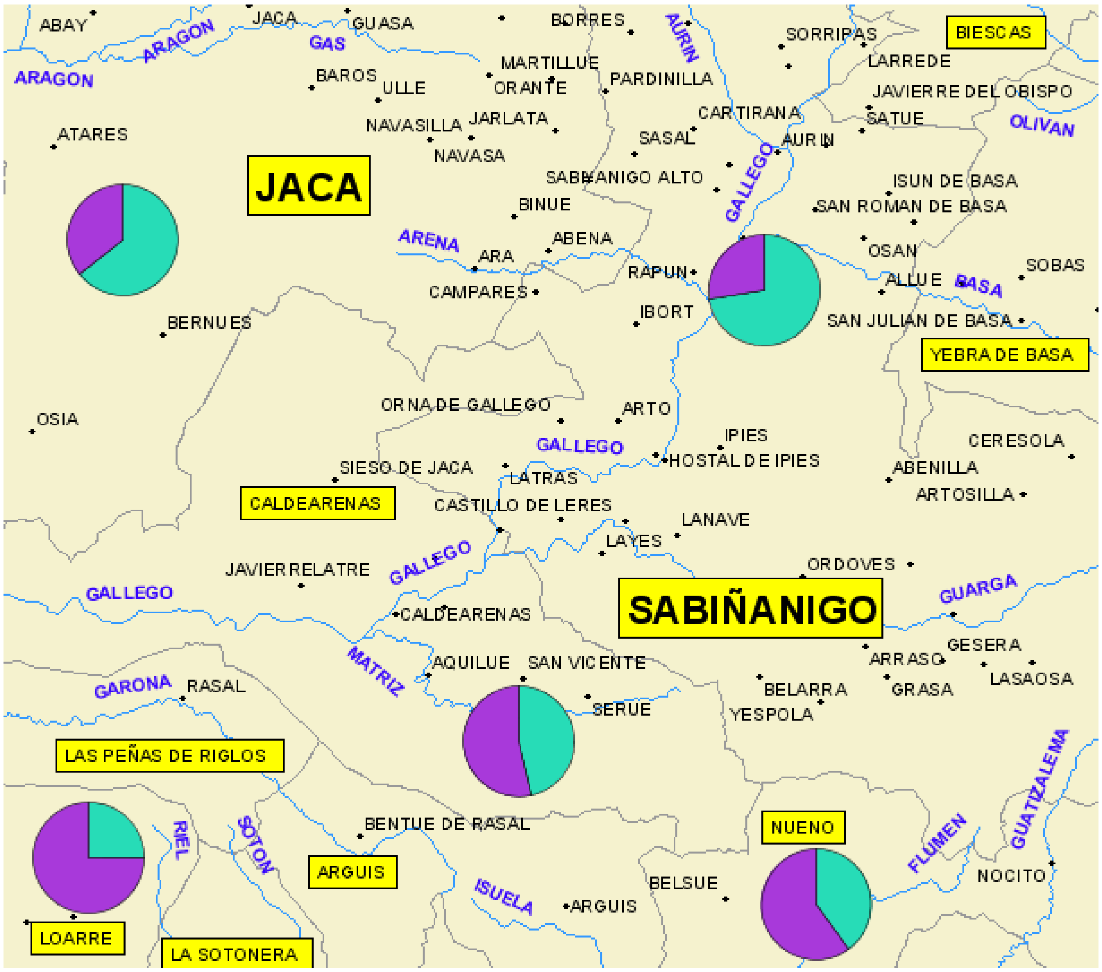

Figure 4.8: Sample Map with Texts and Charts

this class may be instantiated only once for a JMapControl, and holds all the labels in that map. LabelLayer plays the role of the Map in the pattern. A LabelLayer may have a LabelRenderer, actually several but this is not shown in the diagram, which holds all the information about the LabelSymbols used to depict each Label. LabelSymbol plays the role of LabelRenderer in the pattern. LabelSymbol class in JGISView is extended to allow for very different labels, from plain text to statistical charts, making thus profit of the Generic Label pattern. LabelSymbols produce ScreenLabels, which play the role of the GenericLabels in the pattern. Each ScreenLabel has an instance of Label, the class which plays the role of LabelContent in the pattern. Each Label points to the data that is going to be used to draw it; the data are in an instance of the class Fields, which hold the attribute information associated with a feature in a MapLayer of the JMapControl (although in JGISView there are not features as such: JGISView uses Recordsets where each record (Fields class) is a set of attributes that would correspond to a feature).

4.4 JMapServer: a Web Map Service Implementation in Java

JMapServer is an OGC Web Map Service implementation, see 4.1.3, developed in Java, and built around the geographical data management and rendering capabilities of JGISView, presented in 4.2.

Around the rendering capabilities of JGISView, JMapServer adds the functionality needed to support the OGC interfaces and the configuration of the WMS. This is shown in figure 4.10: the main component, named JMapServer, integrates a component, the Map Request Builder, that receives map requests and translates them to a format that JGISView can understand to render those maps. To do that, it uses the Capabilities Manager, which is the component in charge of relating the maps, layers and styles offered by the WMS with those understandable by JGISView (i.e. map definitions stored in JGISView project files). The Map Request Builder then generates an image file, in one of the format that the WMS supports, and returns it. The JMapServer component has an RMI interface created following the design of the OGC WMS standard (see [137]

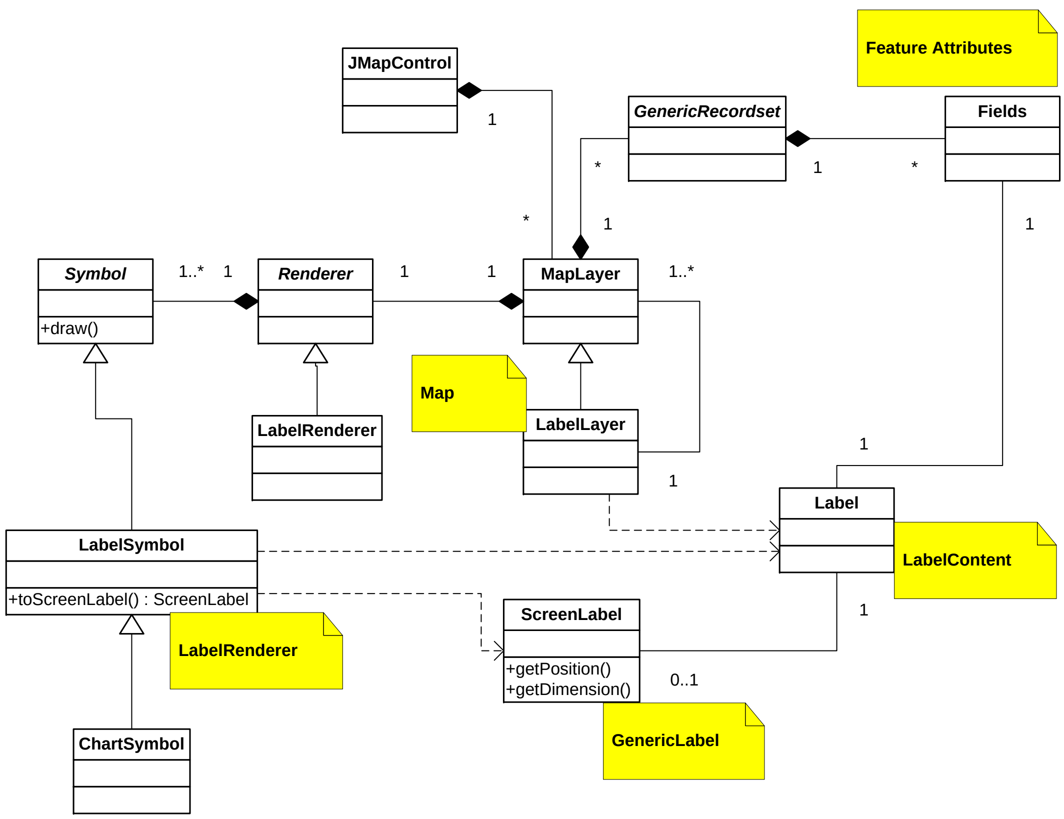

Figure 4.9: Generic Labels in JGISView

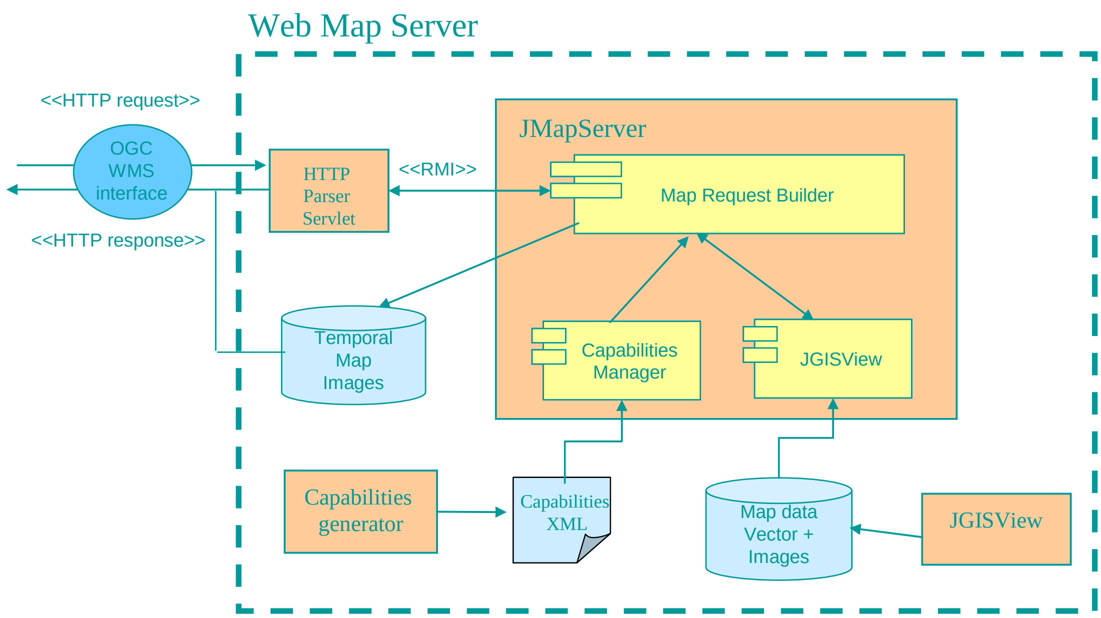

Figure 4.10: JMapServer Architecture

for a good introduction to RMI).

The JMapServer component generates maps following the requests, but the OGC specification requires HTTP interfaces. An extra component is thus needed to parse the HTTP requests received by the Web server. A Java servlet [159] is used for translating the HTTP requests, following the OGC specified format, to the RMI requests understandable by the JMapServer component, and then for translating the responses too. This design allows to separate the functionality offered by JMapServer and the OGC interfaces making it easier, for instance, to support a different Web service interface, maybe based in SOAP. Besides this, the JMapServer component can be accessed through its RMI interface by other Java programs in a more convenient way.

JMapServer classes are designed to fulfill the WMS specification, and thus guided for its operations. Figure 4.11 shows the main classes and interfaces that allow for this. The MapServer interface has three methods, getMap(), getFeatureInfo() and getCapabilities(), which correspond to the operations in a WMS. The JMapServer class is in charge of processing the requests, creating responses and returning the data generated (images for getMap requests, text information for getFeatureInfo requests and XML for getCapabilities requests). The RemoteMapServer class provides an RMI interface with similar operations to those in the WMS specification. It acts as an RMI proxy to an instance of MapServer. The HTTPMapServer class, implemented as a Java servlet, provides the mandatory HTTP interfaces to fulfill the WMS specification. This class acts as an HTTP proxy to a RemoteMapServer.

JMapServer integration with JGISView is also shown in figure 4.11. Class WebServerMapControl extends JMapControl, the main class in JGISView, with the method exportMapAsImage() and implements two interfaces, MapProvider and FeatureInfoProvider. A WebServerMapControl will then be registered with the JMapServer to fulfill getMap and getFeatureInfo requests. A DefaultCapabilitiesProvider class fulfills the third operation of the MapServer interface, getCapabilites. Layers and styles of an OGC WMS are mapped to JGISView Layers: an OGC style can be composed of many JGISView Layers, what gives us a lot of flexibility to configure them.

4.5 Examples of JGISView and JMapServer in Use

This section presents some applications created using JGISView and JMapServer. These applications make profit from the standard specifications for GIS Web services promoted by the OGC and demonstrate how these specifications are a suitable approach to modularize many GIS applications.

4.5.1 An Olive Tree Recognition Application

Many European countries, including Spain, have to check the number of olive trees declared by farmers against reliable data, in order to provide them with subventions from the European Union. This section shows the use of a Web services architecture

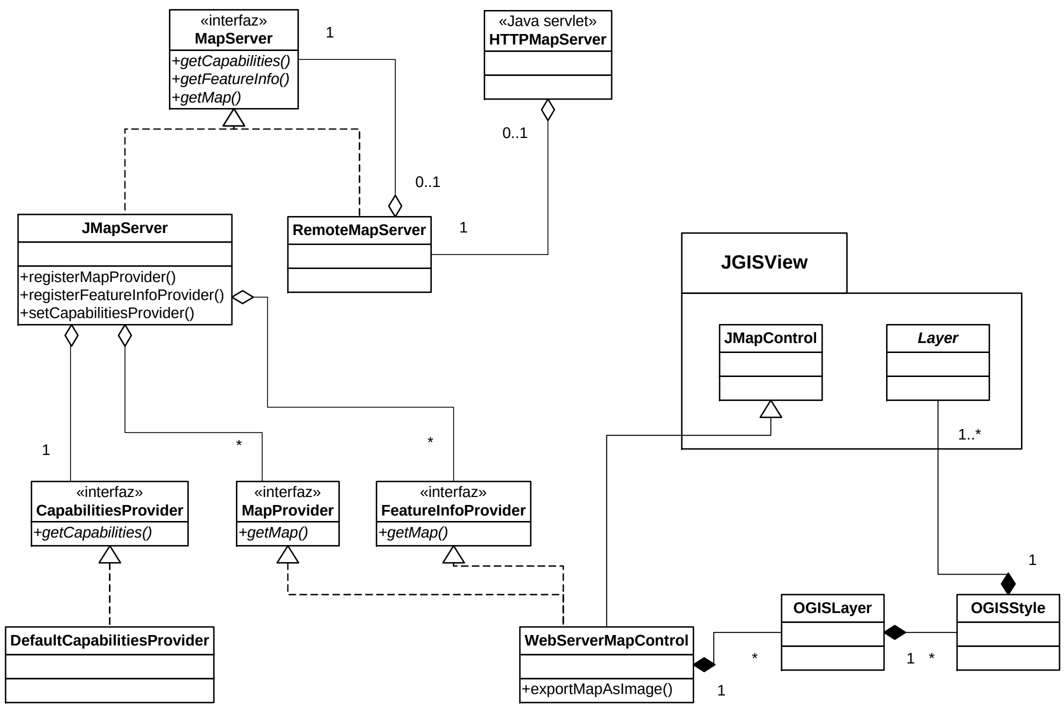

Figure 4.11: JMapServer Design

in the development of a GIS application that supports a computer-assisted olive tree counting on aerial images. This application has been developed for the Agriculture Department of the Spanish region of Aragon.

European Union subsidies to loss-making agriculture areas allow the growth of high quality and low price competitive products. The policy taken by the European Union, regarding with olive trees farmers subsidies, is that the amount of money received by these farmers is based on the number of olive trees they own. The numbers calculated by the administration are collated with those declared by the farmers to ensure right funds allocation. Currently, this checking usually requires in situ inspections carried out by government personnel. To reduce this effort, solutions based in remote sensed data, for example from aerial images, are increasingly being adopted.

There are five countries involved in olive trees subsidies: Spain, Italy, Greece, Portugal and France. In order to assist the olive tree register creation in Portugal and Greece, the Joint Research Center (JRC) of the European Commission developed a GIS tool that enable the computer-assisted counting of olive trees on scanned aerial photography. The work done in this project mainly consists in the development of an algorithm for tree recognition in digitized aerial photography and its integration with a GIS environment (in this case, ESRI ArcView) [139, 68].

The Spanish administration adopted the solution of creating a Web GIS application named ‘SIG Oleicola’. This application supports the automatic detection of olive trees, and the visualization and management of the geographical information needed in the tree identification process.

In both cases the solutions involve ad-hoc applications that only provide local data access and do not provide with a good interoperability. This makes difficult communicating with other software components, integrating new geographic information (i.e. more accurate aerial images or other cadastral raster information), or adding new ways to manage the information linked with the olive tree register.

In the emerging open and distributed environments, interoperability is essential for many systems, including GIS-based applications. The efforts of the OGC have led to several specifications that describe a technical infrastructure for the interoperability of

GIS systems [174]. Three of the fundamental georeferenced information access services are the Web Map Server (WMS), the Web Coverage Server (WCS) and the Web Feature Server (WFS).

The Web Map Server (WMS) is described in section 4.1.3. The Web Feature Server (WFS) [171] and the Web Coverage Server (WCS) [59] provide access to collections of data through well-known interfaces. The WFS delivers GML [48] representations of geospatial features in response to HTTP queries. Clients access geographic features through a WFS by submitting requests for just those features they need. A WCS supports the interchange of raster geospatial data in different formats. Different services can be chained to support information production workflows.

Besides the automatic olive trees recognition, the visualization of geographical information useful in the identification process is another requirement of the system. The geographic information used by this system can be divided into three groups:

- Vector information: groups all the digitized vector cadastral information that represents the parcel bounds and other additional information about parcels, and the geographical elements that surround them. This information is not available for all the municipalities in the region of Aragón.

- Raster information: is made up of the ortho-photographs, with a 1 meter resolution, used as basic cartography, and the scanned maps needed for the areas where vector information has not been created yet.

- Alphanumeric data: this information includes values related with olive trees (number of trees, geographic coordinates. . .).

In order to overcome the limitations that similar systems show, an architecture based in Web services was designed, with the OGC compliant components described before, as shown in figure 4.12. The vector cadastral information is accessed by means of a WFS that delivers the requested features (parcel lines, parcel centers, . . .) for a specific geographical extent, or associated to a cadastral identifier (numeric). Another WFS is used to retrieve the olive tree locations from the existent official register. A WMS is responsible for providing the graphical representation of the ortho-photographs

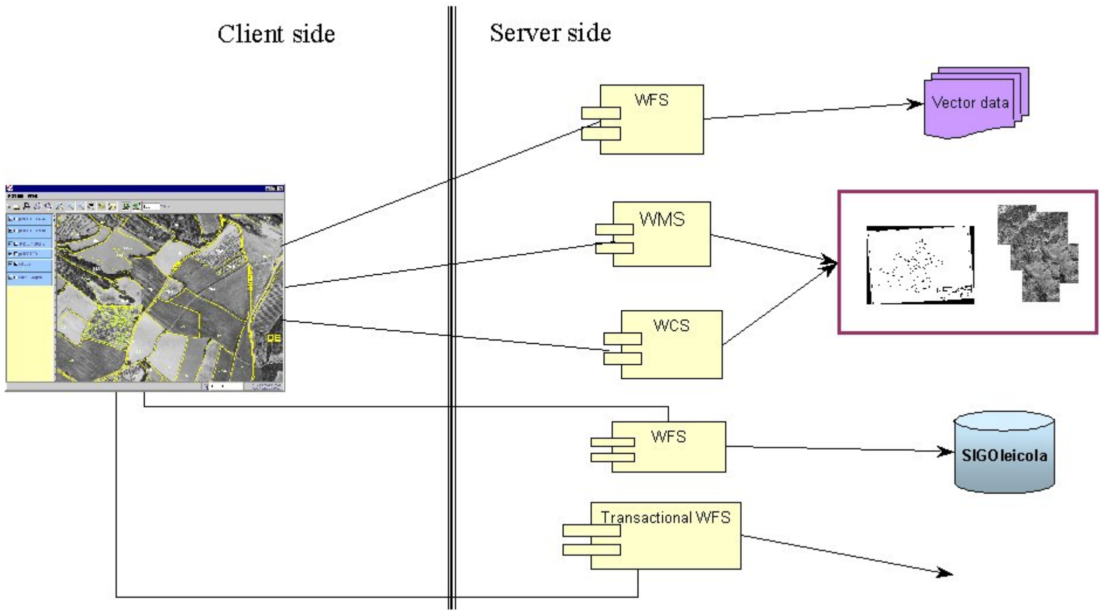

Figure 4.12: Olive trees recognition system architecture

and the raster cadastral data. An interaction with a coverage server is needed when a scanned map is requested. The WCS provides the extent of the desired scanned map, since this information is required to make the following requests to the WMS.

Using these standard components allows for future uses of the services they provide by other software components, by means of their well-known interfaces. This is important considering the fact that the services offered may be generic enough to be useful for other entities or administrations or for applications solving different problems. The development of a component based GIS may also facilitate the design of customer specific interfaces [142], i.e. Web clients, or remote applications, giving the same functionality with different end-user interfaces. Another important characteristic of this application is its extensibility, that allows for the easy inclusion of other geographic data.

The main use of this information system is the visualization of the cadastral information with all the information linked to the register of olive trees. The main graphical interface of the application is shown in figure 4.13. A cadastral municipality is formed

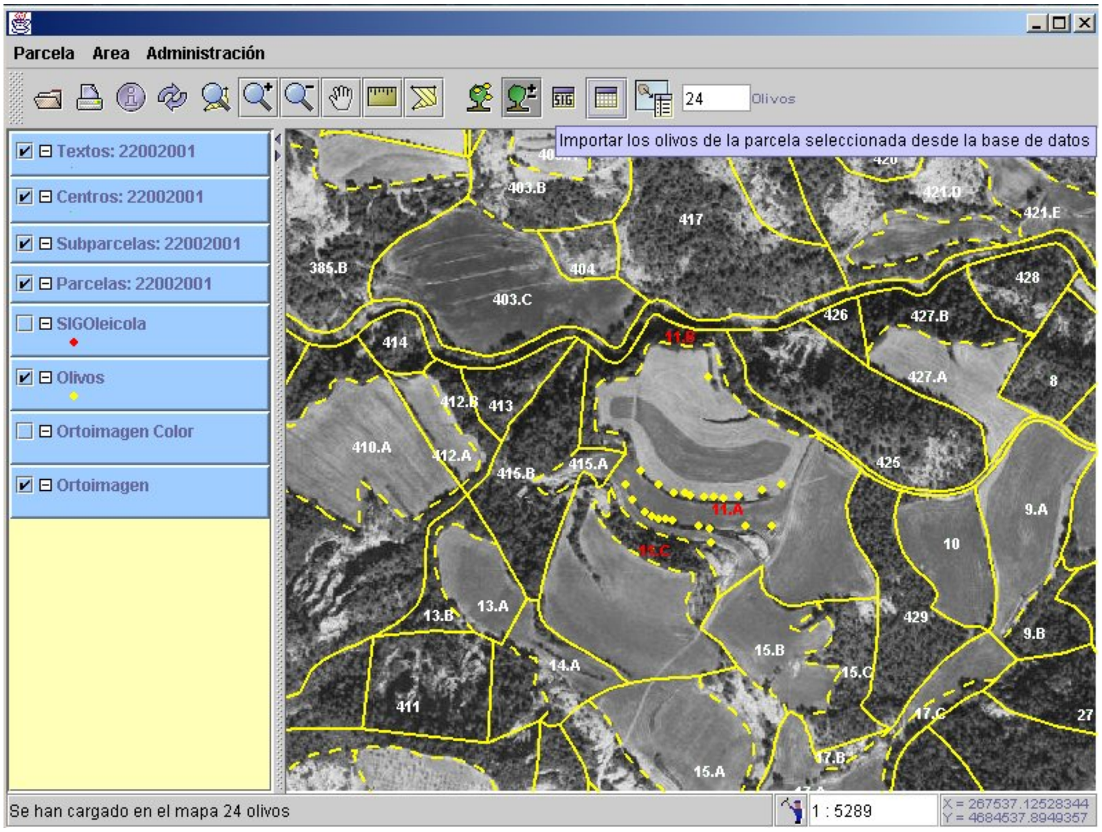

Figure 4.13: Olive trees application main interface

by polygons. Each of these is divided into one or more parcels. The two geographical entities that lead the users workflow are these polygons and parcels. The users selects the cadastral unit that they desire to visualize and the application makes the required requests to the right servers, depending on its type (vector or raster). If the unit selected is vector type, the application shows the polygon and parcel bounds, the subparcel bounds if they exist, the parcel center and some important geographic features as rivers, roads or villages. If the vector information for the parcel or polygon requested is not available, the corresponding scanned map is used Each of these maps represents a polygon, except when the polygon is large enough to be represented in several maps.

Previous olive trees registered locations can be visualized. These positions can be modified carrying out a process to detect them on the image. The resolution of the ortho-photographs (a pixel size of 1 meter) allows the application of computer vision techniques to identify olive trees on the image. This detection process is interactive, allowing the user to improve the results manually, and requires user supervision to obtain the most suitable parameters for a certain zone, since the olive trees could vary in shape, size or colour between zones.

4.5.2 Serving Climatic Data Series on the Internet

Climatic information is a kind of geographical information useful in many different interest areas (agriculture, environment, tourism, public health...). This data is interesting by itself (i.e. weather information), integrated with other kinds of geographical data (i.e. for more detailed tourist maps, or natural resources management [124]) or as input to different predicting models (i.e. epidemiology [54]). Climatic data is naturally suited to be represented in temporal series, as it varies constantly. This data is most useful in temporal series in order to develop models and make predictions, as seen for example in epidemiology [54], public health [65], or climatic change studies [173]. Publishing this kind of information on Internet is thus relevant to many researchers in different areas. Many efforts have been made to publish climatic data on Internet, through different map servers or by means of simple coverages download (see as several among many examples [115], [164]). Efforts need to be done in order to increase the accessibility of this information for different kinds of uses and users.

Temporal series of Climatic Data

Satellite images are taken periodically, processed, analyzed, stored and prepared for visualization, access and thematic maps production [121]. An enormous quantity of information is generated. This information is prepared for its integration in GIS as coverages, layers, for different formats and computer systems. When the same data are taken periodically on the same geographical area, the result is a temporal series. Temporal series are a particular case of layer aggregations, where a layer aggregation is a set of layers that share many metadata, and where the main differences in their metadata may be considered parameters (time or date in temporal series of the same region, sensor type in satellite raster coverages of the same area, etc.). Treating a set of satellite images as a layer aggregation, in a GIS context, makes it possible to take advantage of their common metadata, while treating it as a set of unrelated layers would have at least two drawbacks to give access to them:

- Describing the common metadata in every layer in an aggregation is a waste of space, or a waste of bandwidth in a network environment.

- For the user would be more convenient to know that several layers are in fact a layer aggregation because this way would be able to select the parameters of his/her interest instead of having to browse all the layers as if they were different. Analysis can also be more convenient if the software is aware of the fact that layers are an aggregation.

Climatic data in several areas is most useful in time series, in order to make temporal analysis and proper predictions. Agriculture and public health are two good examples:

- Agriculture and products from natural resources are subject to the vagaries of weather (the manifestation of fast atmospheric hydrologic processes) and climate (the long-term statistical measures of these hydrological processes). Thus, even relatively small changes in weather and climate could potentially have important consequences. Observation and monitoring of temperature, rainfall, and other environmental conditions over both short and long time scales are critical tools for making effective land use and resource management decisions.

- Public health practice needs timely information on the course of disease and other health events to implement appropriate actions. Most epidemiological data have a location and time reference. Knowledge of the new information offered by spatial and temporal analysis will increase the potential for public health action. Geographic information systems (GIS) are an innovative technology ideal for generating this type of information [65].

Internet is the best place to offer geographical information like climatic data because, as it is constantly changing, an easily updateable and accessible place is needed. As shown before, climatic data is most useful in temporal series, so the problem of giving access to climatic data time series on Internet must be addressed. There are several solutions for different kinds of users and user needs.

Temporal Series on the Web: Simple Data Exploration

In order to show simple maps on the Web much software is not needed. Publishing a simple map image with a legend does not require from any special server side software. It is not very useful though. Users cannot browse the data in any way or make any kind of queries. Exploration of the data is thus not possible. To offer these exploration capabilities, the existence of some kind of web map server is a requirement. Thus, there are several configurations and combinations of clients and servers with an increasing degree of utility for the users:

1. Fixed maps produced off-line and published as simple static Web pages. No special Internet software is required but a Web map server can be used. In fact using one would make updates and changes much more simple than having to generate the map images again for every change. Useful as simple ways to show a fixed zone, in a fixed moment of time. Users don't need to be able to do anything besides been able to view a Web page. For example, a weather forecast for the current date can be provided with this simple configuration. The temporality of the data is revealed through simple periodical updates of the maps.

2. Maps of a fixed region of interest but with the capability to select dates, or times, and maybe with a limited thematic selection. This solution allows for temporal series of data, including historical data, or even predictions about the future, while maintaining great simplicity. A Web map server is not required, but may be used given the different types of use it offers through the use of clients of different complexity. Again using one would make easier updates. This interactive solution can be used for offering weather forecasts of several days ahead or when historical

climatic information about a determinate area is needed (i.e., studying climatic change of a fixed area).

3. Interactive Web maps offering layer and dates selection and tools for browsing the maps (zooming, panning, etc.). These maps are more customizable by the users because they have more choices, layers shown, area selected, date, etc., but more complex to use too. Offering the option of printing a map once customized by the user gives more possibilities to this solution. Some kind of web map server is needed to implement this. The client software may be a more or less simple HTML page, depending on the options given to the users. A more complex solution for the clients could be based in Java, giving a Java applet to the users with similar options to those found in desktop applications , like this one developed using JGISView (figure 4.14. The client does not need to see all the capabilities offered by the map server. This way the same map server can be used for different needs, with customized client software adequate for distinct necessities. Maps offering data at different detail levels need to offer this kind of interactive access for allowing users to select the region, dates and themes (layers) of interest for them. This solution can be applied when different climatic parameters are of interest and they are presented in several layers. For example, for epidemiological models where temperature and humidity are inputs [54].

4. One step beyond the last solution would be providing access to different map servers with different content on the same map. Interoperation between map servers would allow for data being developed, maintained and changed in different places (they may be different departments in a company or different places in the world) but shown together in the same map. This would allow for adding extra value to existing maps by allowing to overlap on them information from other places. Climatic data for example is very useful in areas so different as agriculture, biology or tourism. Maps providing this kind of information would benefit from being able to connect to a server with climatic data of their areas of interest and show the climatic information on top of the maps they already provide. If they already serve temporal data (i.e. annual crops for the last years),

they can benefit from the temporal series of climatic data offered by the climatic servers they are interoperating with, to highlight trends and relationships hidden before for not having had access to that specific climatic data.

Temporal Series on the Web: Advanced Data Exploration

There are users with more specialized needs. They may need to fully access to the capabilities of the map server, in order to know the different coordinate systems that they can use to retrieve the maps, all of the layers and styles offered by the map server in order to compose the maps exactly as they need them or the full range of the different parameters needed to retrieve certain layers. They may be interested for example in maps showing the raw data (i.e. radiometer values) used for calculating some climatic parameters (i.e. temperature) before they have been processed and composed for visualization. Other users may even need to download the complete coverages to work locally. The interactive mapping tools offered for browsing the data in the map server are thus used as a means of exploring these data to find and download exactly the needed information:

5. Any client able to connect to a Web map server can be used as the tool for this kind of user, from Java applets embedded in Web pages to complete desktop applications. This kind of software should be able to offer the user the possibility to see the capabilities published by the selected map server, provided that the map server publishes them, in order to find out all the metadata they may need to know, such as coordinate systems of the layers, all the layers and styles offered by the map server, data about the maintainer of the coverages etc. Other interesting option is the ability to connect to different map servers selected by the user, provided that he knows their URL and interoperation is possible. This allows for composing maps exactly as the user desires and makes it possible to find new ways to put together different kinds of geographic data offered by different map servers. It would allow for the same type of uses than option 4, but with more precise geographical tools.

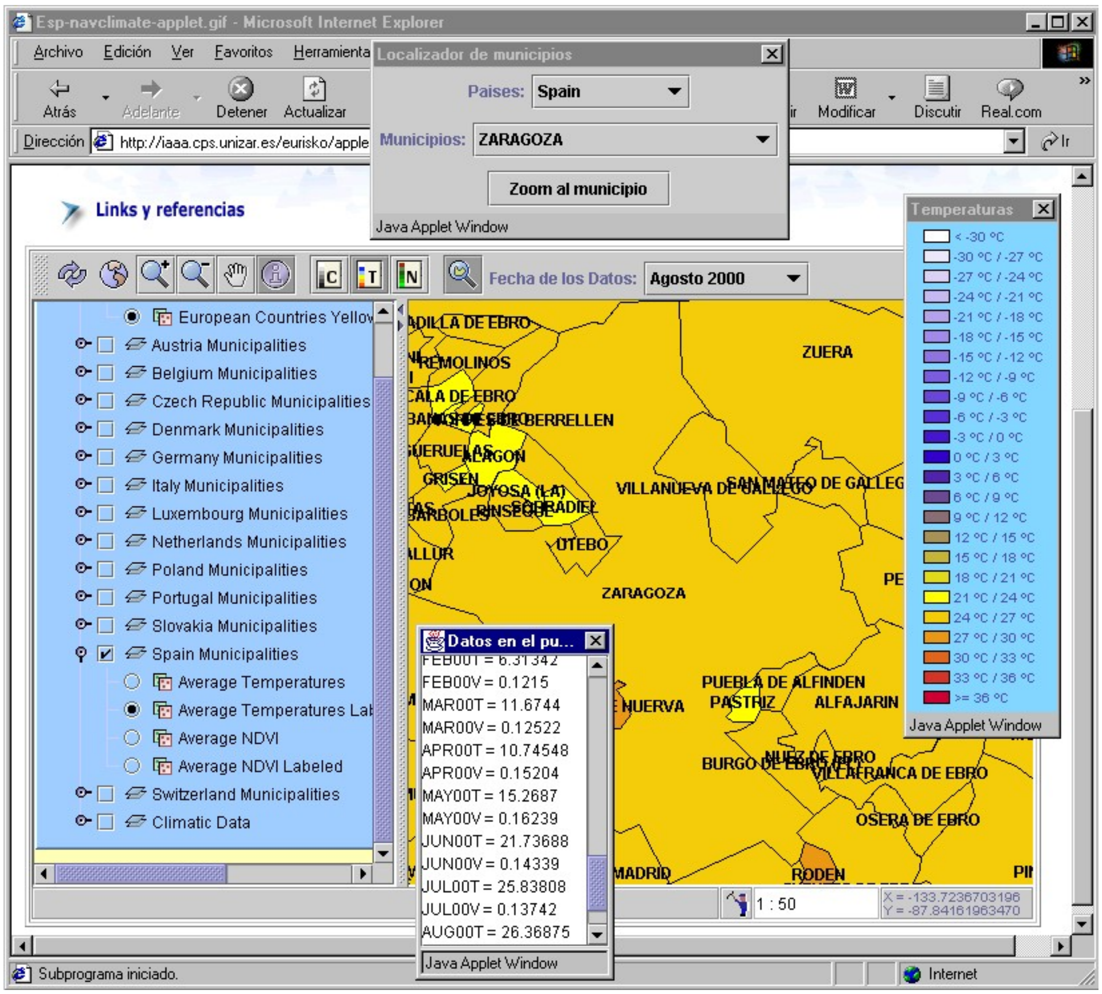

Figure 4.14: An example of an advanced climatic map server client

6. Servers may also be extended with new functionality. One of interest would be allowing for downloading the coverages the user has selected and browsed by means of the standard interfaces of the map server. A customized client can exploit the fact that browsing the map is already in the functionality of the generic client software, and add the options needed to download the coverages the user is seeing after browsing the map server. This way climatic data series can be used locally by researchers in the fields, listed before, that need them.

Temporal Series on the Web: Data Exploitation

If the Web map server is flexible in what relates to graphic formats offered, it may be used as a very limited coverage server. Asking this server just one of its raster layers in a graphic format suited to the needs of the user, the different layers in the server may be accessed through the standard interfaces:

7. A desktop application able to exploit this possibility could benefit from the advantages offered by working directly with data in the server. When coverages are updated in the server, the client automatically works with the new data. There is no need to notify the clients the change or to make them download the new coverages. Temporality of the climatic data makes this option very convenient, as users can access to the latest information as it is being made accessible. The same map server can be used for browsing through the coverages the client does not need complete, reducing thus the traffic from the server and increasing the speed, and for downloading the coverages the client does need. Users can work from different places or computers because data is stored in the Web. A user does not need to carry the coverages from one place to another or from one computer to another. Provided that the computer is connected to Internet, data is always there. Remote and local coverages can be managed together in the same way, because the desktop application can access both. Climatic data can be integrated with local models to improve them, or the check the validity of their predictions. There is no need to download an entire, potentially very large, coverage if the user just uses one geographical area. Figure 4.15 illustrates this scenario.

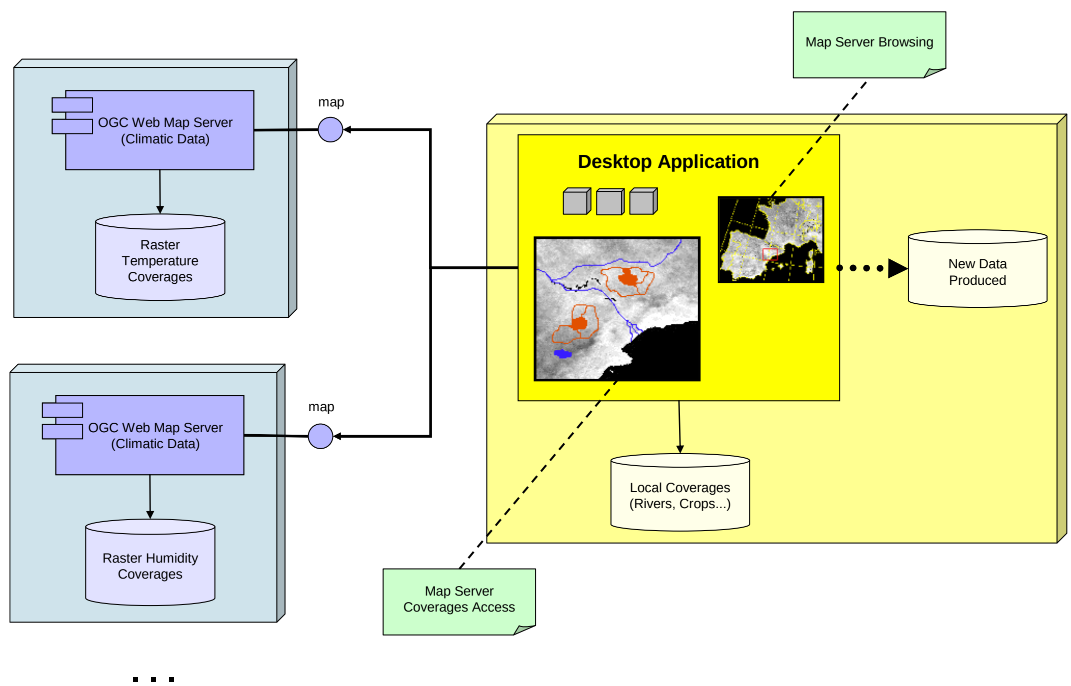

Figure 4.15: Data Exploitation Through Interoperable Web Map Servers

	Show map	Parameter (date / time) selection	Map navigation	Several data sources (map servers)	User selection of data sources	Coverage downloads	Work with local data	Map Server required?
1	Yes							No. May be used
2	Yes	Yes						No. Should be used
3	Yes	Yes	Yes					Yes
4	Yes	Yes	Yes	Yes				Yes
5	Yes	Yes	Yes	Yes	Yes			Yes
6	Yes	Yes	Yes	Yes	Yes	With a proper extension		Yes
7	Yes	Yes	Yes	Yes	Yes	On the fly, not stored	Yes	Yes

Table 4.1: Climatic data Web access summary

Temporal Series on the Web: Summary

The table 4.1 shows a summary of the characteristics of the different options described before to give access to climatic data time series.

Serving Climatic Data Series with JMapServer

Offering the temporal series of climatic data through a standard Web map server, with well-known interfaces, makes it possible to interoperate with other systems (platform-level interoperability according to [179]), allowing for different uses of the information, from simple map visualization to direct access to the layers from complex desktop applications.

A Web map server compliant with OGC specification (as JMapServer is) can be used to provide very different levels of access to time series of climatic data, exploiting the interoperability possibilities offered by its well-known interfaces and the option of developing client software specifically suited to different user needs, as these may impose several constraints or requirements on the software [36]. Such a map server offers most of what is needed for the most advanced ways of serving temporal series of climatic data described before:

- Interoperability through its well-known standard interfaces (option 4).

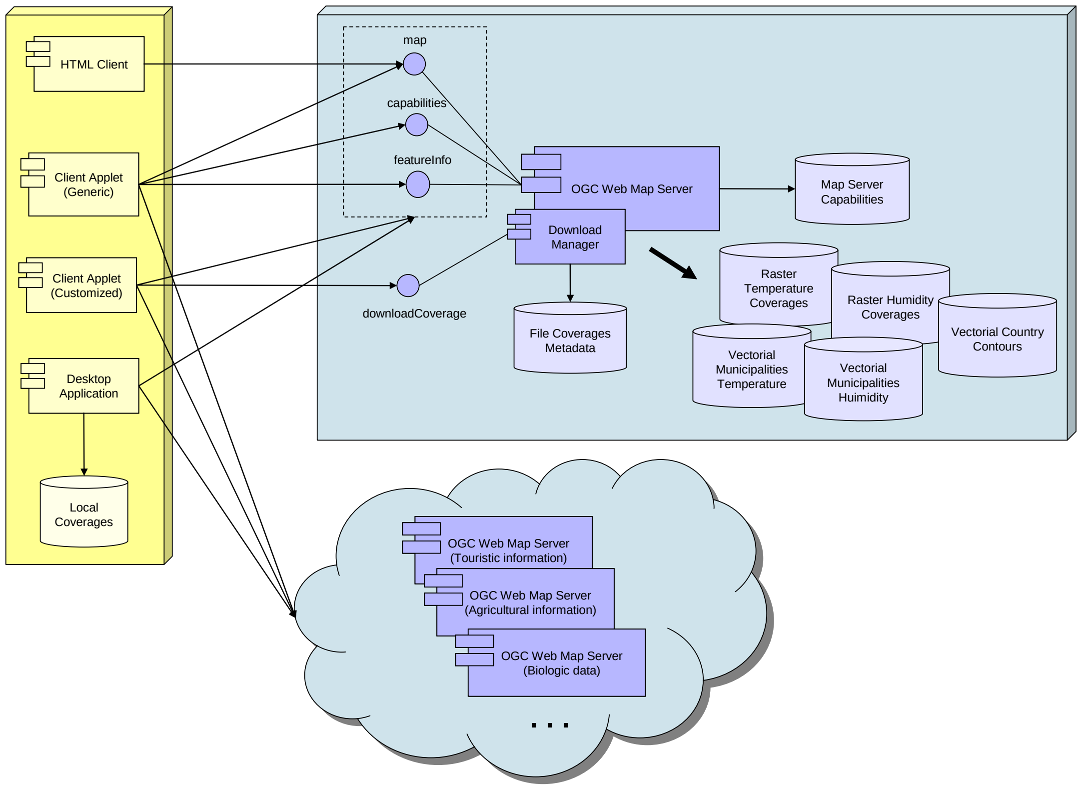

Figure 4.16: Climatic OGC Web Map Servers Architecture

- It makes public its capabilities thus allowing users to know them (option 5).
- It is flexible in what relates to graphic formats offered (data exploitation).
- It is extensible through the 'vendor specific capabilities' (option 6).

Figure 4.16 shows an example of architecture able to support all the different options described in the previous chapter by interoperation between several kinds of clients and OGC compliant Web map servers.

4.6 Conclusions

In this chapter, several contributions to the modelling of SDI portrayal services and applications that use them have been presented.

JGISView, a Java component designed to support basic GIS visualization capabilities, provides the connecting thread of this chapter. The design model presented in section 4.2 shows the core classes of this tool. Several of its main GUI classes were designed to follow the JavaBeans component model to facilitate the design of GIS desktop applications that use JGISView.

The cartographic quality of the maps produced by JGISView is important. Simulated annealing is a well-known technique to support good quality labelling of maps. In order to apply this technique to support different types of labels, their fundamental characteristics were abstracted: a label was considered as any data item that can be portrayed on a map, in a certain position and with a certain bounding polygon. This abstraction was the base to propose the analysis pattern to support the modelling of generic labels in GIS applications presented in section 4.3. This pattern separates the generic labels from the labelling algorithm, allowing thus to implement generic labels in different GIS applications, that may have different labelling algorithms already implemented. Although the map labeling problem has been addressed in the bibliography, results have been typically presented as algorithms. The object-oriented approach, the dominant paradigm in software for many years, requires more than algorithms to specify solutions for software problems. This work has thus addressed the map labeling problem the 'object-oriented way', by offering a pattern that can be easily applied to other software with similar requirements. Although simulated annealing has been chosen for the implementation of this work, many algorithmic contributions to the map labeling problem can easily be adapted for its use with generic labels, as they abstract most of what it is needed in order to place labels (position, and bounding area). The quality and efficiency of the solution are mainly given by our implementation of simulated annealing, an algorithm already well-known, tested and used by commercial software, but they do not depend on, and thus are not limited by, the generic label pattern.

JMapServer, a Java implementation of an OGC Web Map Service, is also discussed in this chapter. JMapServer architecture includes JGISView as the rendering component, what allows its users to employ the graphic interface of JGISView to create the same maps that are going to be served by JMapServer.

JMapServer has been used to test the feasibility of the WMS specification in real applications. Section 4.5.1 presents a real example that uses JGISView as GIS component in a desktop application acting as a client of JMapServer. In that section, the advantages of using a standard Web services based architecture in the development of a GIS tool have been presented. This architecture leads not only to the fulfillment of the user requirements, but also to the design of an open, standard system, able to interoperate with other systems. Through the case of the olive trees identification system we have been able to show more clearly the advantages of this interoperability. Other solutions previously developed, only provide the user functional requirements, whereas this application adds the ability of providing or requesting services to / from other components without architectural changes. As an example, the architecture presented here has been reused to create a thin Web client that provides the remote visualization of the cadastral information involved in the application. In the future, new services and new geographical information will be added to the system and this system will also need to request information from other standard servers. The chosen architecture will allow for these improvements without major changes, and will also continue to offer these new services and data to other software components from other departments by means of its standard well-known interfaces.

In section 4.5.2 seven scenarios have been listed, showing different levels of complexity in the access to temporal series of climatic data, with different tools for the different needs of users from curious people to serious researchers. The use of interoperable Web map servers to give access to this information has been proposed as a means to increase accessibility and as a means to provide all the tools the different scenarios need. The use of Web map servers compliant with the standard Web map service proposed by the OGC is proposed as a solution, because they are adequately suited to solve most of the problems presented before and are flexible enough to support extensions for giving more options. A sample architecture showing how the different scenarios could be addressed is outlined, and several of them have been implemented with JGISView and JMapServer as client and server respectively.

JGISView and JMapServer have also been used in several other projects [108, 4].

Chapter 5

Conclusions and Future Work

5.1 Research Contributions

As shown in chapter 1, the most common definitions for the term Spatial Data Infrastructure refer to the technologies, policies and institutional arrangements which have the objective of facilitating availability and access to spatial data, and services. These definitions put SDI under the umbrella provided by Information Infrastructures, a term that encompasses large, distributed, shared and enabling information systems with both social and technical components. In chapter 2, this thesis analyzes SDIs from this point of view, modelling them as federations of autonomous communities.

The technical components in SDIs can be traced to geographic information systems, digital libraries and service oriented architectures. In chapter 3, this work focuses on the technical components of the communities that compose an SDI, studying the component based information system architecture they need to be part of that SDI.

ISO TC/211 and the OGC have provided specifications and standards which allow the technical components of different organizations, that are independently managed, to interoperate. In chapter 4, different aspects of the design of one of these component types, the portrayal service, are addressed.

As a summary of the results of this PhD thesis, these are its main contributions:

- In chapter 1, in order to provide a context for this thesis, the foundations of SDIs are presented: Digital Libraries, Service Oriented Architectures and Information Infrastructures are reviewed in relation with SDIs. Systems of Systems are proposed as a previously unconsidered conceptual framework for SDIs. Several aspects related with the different software engineering methods applied in this work are also presented. Part of its contents are included in [29, 28].

- In chapter 2 this thesis proposes a software systems approach to several characteristics of SDIs from an Information Infrastructure point of view, by providing a model following the ISO RM-ODP Enterprise Language. This model includes the main components in an SDI, both social (people, organizations, policies...), and technical (software components, datasets...), and may be tailored to design or document different SDIs. Another benefit of this model is that it provides an indirect definition of SDI: a federation of communities built to facilitate and promote the use of spatial information resources, on a stable and supporting environment, in a certain extent where different autonomous relevant organizations coexist, and where it is desirable, or necessary, to keep some of that autonomy. Most of this chapter is included in [27]; other related published contributions are [23, 180].

- In chapter 3 this thesis systematizes, refines and extends several proposals found in the literature for the software component architecture of SDIs. This is presented as an architectural style, under the Component & Connector viewpoint of the 'Views and Beyond' methodology. The style extends two well-known ones, the Client-Server and the Shared-Data, with domain knowledge related to SDIs, providing thus a tool and a shared vocabulary to help SDI architects to design and document the technical components of these infrastructures. This chapter contents have been published in [18, 24, 26]. Other previous works that contributed to shape the results in that chapter are [163, 105].

- In chapter 4 this thesis has proposed several contributions regarding the portrayal services of SDIs, i.e. those related with the cartographic visualization of spatial data. A Java GIS visualization component, JGISView, is the connecting thread of these contributions. Several aspects of its design and its integration in JMapServer, a standards compatible Web Map Service, are presented as strategies to provide GIS visualization capabilities to different types of applications. These strategies show the feasibility of the standardized interfaces approach found in SDIs. An analysis pattern to support different types of labels in GIS software is also presented. This pattern separates the labels from the labelling algorithm, making it easy to implement new types of labels, or new labelling algorithms, in a GIS software. Most of the contents of this chapter have already been published [62, 21, 25, 20, 8].

- Finally, it must be highlighted that several software components related with this thesis are included in several programs registered in Spain Intellectual Property General Register (*Registro General de la Propiedad Intelectual*) [184, 188, 183, 187, 185, 182, 186].

5.2 Future Work

Although the term information infrastructure is not clearly defined, the relationship between SDIs and IIs has been established in the literature almost since the conception of the former. This relationship provides a conceptual framework to advance in the knowledge about SDIs that has been already used in the SDI literature. Nevertheless, there are other possible frameworks for SDIs. The systems research community has been working on the concept of system of systems, a complex system that is composed of several independent systems, for the last few decades. Although this concept is neither mature nor well defined, it has been useful to characterize a new class of systems that need specific solutions because they provide us with new challenges.

In this thesis, the relationship between IIs and SoS has been shown in section 1.1. These terms are used to refer to similar concepts, although from different points of view. SoS is a broader term, so IIs can be considered as SoS with several distinctive characteristics: their enabling nature, the focus on information and their dependence on an installed base.

The relationship between IIs and SoS provides a new framework to study SDIs. The research on SoS provides us with knowledge that can be applied to the study of SDIs. SoS engineering and architecture are emerging approaches to address problems that traditional systems engineering and architecture techniques do not solve well. These approaches provide us with tools, processes and methods to model, develop and evolve SDIs.

SoS literature highlights those areas that are in need of research. On the one hand, most of these areas include research problems similar to several of those addressed by the SDI community, and thus of possible interest for its members. The socio-technical nature, the lack of a central authority, i.e. the distributed management and the required coordination that this requires, the necessity to develop, or as some authors have pointed out to 'cultivate', them over a existing technological and social base, or the existence of desirable, or undesirable, emergent properties are among the research problems faced by the SDI community that are common with those of the SoS community. On the other hand, the SoS research community could benefit from examples and solutions provided by the SDI community and expressed in their terms.

SoS architecting is a relatively new field, and thus it presents many research challenges. Many of these challenges could be considered by the SDI research community, not only to find solutions applicable to SDI development, but to extend solutions and studies created for SDIs to apply them to generic SoS. A workshop celebrated at the University of Southern California intended to establish a research agenda for SoSA [167]. Among the items in this agenda, several of them can be interesting for the SDI community because they address SoS characteristics, existing or desirable, which are common with, or also desirable for, SDIs: resilience, an attribute of a system that makes it less likely to fail and more likely to recover from major disruptions; human limits to handling complexity; net-centric vulnerability, highly interconnected systems show vulnerabilities because of these connections; evolution of the SoS; guided emergence, similar to the 'cultivated approach' in IIs and SDIs; or the study of systems without a single owner. Other proposed research items, system versus SoS attributes, model driven architecting and multiple SoS architectural views, are related to the improvement of architectural techniques to address complex SoS. Finally, there is one research challenge suggested for SoS, illustration of success, that could benefit from examples taken from the SDI community.

Bibliography

[1] Alain Abran, James W. Moore (executive eds.); Pierre Bourque, Robert Dupuis, Leonard L. Tripp (eds.). *Guide to the software engineering body of knowledge: 2004 version.* IEEE Computer Society, Los Alamitos, California, United States of America, 2004.

[2] Alameh, N. Chaining geographic information Web services. *IEEE Internet Computing 7*, 5 (2003), 22–29.

[3] Albert, C., and Brownsword, L. Meeting the challenges of commercial-off-the shelf (COTS) products: the information technology solutions evolution process (itsep). In *First International Conference on COTS-Based Software Systems (ICCBSS 2002)* (2002).

[4] Álvarez, P., Bañares, J., Béjar, R., Blasco, S., and Latre, M. Á. Integration of mapserver capabilities in an automatic vehicle monitoring system to extend access through the Internet. In *Proc. of the 4th AGILE Conference on Geographic Information Science. GI in EUROPE: Integrative, Interoperable, Interactive* (Brno, Czech Republic, 19-21 April 2001), pp. 676–682.

[5] Álvarez, P., Béjar, R., Cantán, Ó., Fernández, P., Latre, M., and Muro-Medrano, P. R. Acceso por intra-internet a grandes repositorios de

imágenes de satélite, aplicación a ortoimágenes de la cuenca del ebro. *Mapping 72* (September 2001), 6–12.

[6] Álvarez-Robles, J. A., Zarazaga-Soria, F. J., Latre, M. Á., Béjar, R., and Muro-Medrano, P. R. Water quality monitoring to support the european commission's water framework directive reporting requeriments. *Transactions in GIS 11*, 6 (December 2007), 835–847. In press.

[7] Ambler, S. W., and Jeffries, R. *Agile Modeling: Effective Practices for Extreme Programming and the Unified Process*, 1st ed. Wiley, 2002.

[8] Andreu, E., Béjar, R., Latre, M. Á., Martínez, S., and Muro-Medrano, P. R. A pattern-based approach to support automatic homogeneous map labeling with texts, charts and other elements in a wms. In *Proc. of the 9th AGILE International Conference on Geographic Information Science: Shaping the future of Geographic Information Science in Europe* (Visegrád, Hungary, April 2006), pp. 283–290.

[9] Annoni, A., Craglia, M., and Smit, P. Comparative analysis of nsdi. In *8th EC-GI&GIS Workshop* (Dublin, Ireland, July 2002).

[10] ANZLIC. Asdi distribution network: The internet framework technical architecture (v2.6). Final draft, January 2003. Retrieved September 2, 2008 from http://www.anzlic.org.au/get/2375015862.pdf.

[11] ANZLIC. Australian Spatial Data Infrastructure. http://www.anzlic.org.au/infrastructure_ASDI.html, 2008. Last Accessed on December 16, 2008.

[12] AST Working Group. Inspire architecture and standards position paper. Tech. rep., INSPIRE Working Groups, 2002.

[13] Atkinson, R., and Box, P. United Nations Spatial Data Infrastructure (UNSDI) proposed technical governance framework. Preliminary report 1.1, United Nations Geographic Information Working Group, March 2008.

[14] Bernard, L., and Craglia, M. SDI - from spatial data infrastructure to service driven infrastructure. In *First Research Workshop on Cross-learning on Spatial Data Infrastructures and Information Infrastructures* (Enschede, the Netherlands, April 2005). Retrieved December 15, 2006 from http://gi-gis.jrc.it/ws/crosslearning/papers/PP Lars Bernard - Max Craglia.pdf.

[15] Bernard, L., Craglia, M., Gould, M., and Kuhn, W. Towards an SDI research agenda. In *11th EC-GIS* (Alghero, Sardinia, Italy, June 2005).

[16] Bernard, L., Einspanier, U., Lutz, M., and Portele, C. Interoperability in gi service chains - the way forward. In *6th AGILE Conference on Geographic Information Science* (Lyon, France, April 2003), pp. 179–187.

[17] Bernard, L., Kanellopoulos, I., Annoni, A., and Smits, P. The European geoportal - one step towards the establishment of a European spatial data infrastructure. *Computers, Environment and Urban Systems 29* (2005), 15–31.

[18] Béjar, R., Aboal, J., Gould, M., Muro-Medrano, P., and Vila, P. Spatial data infrastructure in the Galicia region: An instantiation of the INSPIRE framework. In *Proc. of the 6th AGILE Conference on Geographic Information Science* (Lyon, France, 23-26 April 2003), pp. 267–277.

[19] Béjar, R., Aboal, J., Vila, P., Gould, M., and Muro-Medrano, P. Las infraestructuras de datos espaciales en la gestión forestal. una iniciativa de la Xunta de Galicia. *Montes 79* (November 2005), 26–34.

[20] Béjar, R., Blasco, S., Cantán, O., Estrada-Peña, A., and Fernández, P. Serving climatic data series, an OGC WMS approach to facilitate use by means of interoperability. In *Proc. of the 5th AGILE Conference on Geographic Information Science* (Palma de Mallorca, Spain, April 2002), pp. 575–584.

[21] Béjar, R., Fernández, P., Álvarez, P., Valiño, J., and Muro-Medrano, P. R. *Sistemas de Información Geográfica: Una aproximación desde la Ingeniería del Software y las Bases de Datos*, vol. 2 of *Monografías y Publicaciones. Colección Ingeniería Informática.* Madrid, Spain, 2001, ch. JGISView: JavaBeans para la integración de utilidades de visualización SIG en aplicaciones y applets, pp. 185–190.

[22] Béjar, R., Fernández, P., Latre, M. Á., Muro-Medrano, P. R., Rioja, R., and Zarazaga-Soria, F. J. Interoperation of distributed GIS components in a hydrological data inventory system. In *Proc. of the 4th AGILE Conference on Geographic Information Science. GI in EUROPE: Integrative, Interoperable, Interactive* (Brno, Czech Republic, 19-21 April 2001), pp. 685–692.

[23] Béjar, R., Gallardo, P., Gould, M., Muro-Medrano, P. R., Nogueras-Iso, J., and Zarazaga-Soria, F. J. A high level architecture for national SDI: the Spanish case. In *10th European Commission GI&GIS Workshop, ESDI: The State of the Art* (Warsaw, Poland, June 2004).

[24] Béjar, R., Latre, M. Á., Gould, M., and Muro-Medrano, P. R. GIS COTS integration in an SDI software architecture, a study case in the Galicia

region SDI. In *Beiträge zu den Münsteraner GI-Tagen* (Munster, Germany, June 2003), pp. 91–103.

[25] Béjar, R., Latre, M. Á., Monteagudo, S., Muro-Medrano, P., and Navas, M. Geographic information standard web services to improve interoperability: an olive tree recognition application. In *Proc. of the II Jornadas de Sistemas de Información Geográfica (JSIG'02)* (El Escorial (Madrid), Spain, November 2002), pp. 31–38.

[26] Béjar, R., Latre, M. Á., Nogueras-Iso, J., Muro-Medrano, P. R., and Zarazaga-Soria, F. J. An architectural style for spatial data infrastructures. *International Journal of Geographical Information Science* (2008).

[27] Béjar, R., Latre, M. Á., Nogueras-Iso, J., Muro-Medrano, P. R., and Zarazaga-Soria, F. J. Modelling spatial data infrastructures in the enterprise language of the ISO RM-ODP. Submitted to the International Journal of Geographical Information Science, November 2008.

[28] Béjar, R., Latre, M. Á., Nogueras-Iso, J., Muro-Medrano, P. R., and Zarazaga-Soria, F. J. Systems of systems as a conceptual framework for spatial data infrastructures. Submitted to the International Journal of Spatial Data Infrastructures Research, December 2008.

[29] Béjar, R., Nogueras-Iso, J., Latre, M. Á., Muro-Medrano, P. R., and Zarazaga-Soria, F. J. *Handbook of Research on Digital Libraries: Design, Development, and Impact.* Information Science Reference. IGI Global, February 2009, ch. Digital Libraries as a Foundation of Spatial Data Infrastructures. Forthcoming.

[30] Boardman, J., and Sauser, B. System of systems - the meaning of of. In *2006 IEEE/SMC International Conference on System of Systems Engineering* (2006), p. 6 pp.

[31] Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., and Orchard(eds), D. Web Services Architecture. W3C Working Group Note 11 February 2004, W3C, 2004. http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/.

[32] Borgman, C. L. The invisible library: Paradox of the global information infrastructure. *Library trends 51*, 4 (2003), 652–674.

[33] Bowker, G. C., Baker, K. S., Millerand, F., and Ribes, D. *International Handbook of Internet Research*. Springer-Verlag, 2008, ch. Towards Information Infrastructure Studies: Ways of Knowing in a Networked Environment.

[34] Bradstreet, L., Barone, L., and While, L. Map-labelling with a multi-objective evolutionary algorithm. In *Genetic and Evolutionary Computation Conference* (Washington, DC, USA, June 2005), ACM Press, pp. 1937–1944.

[35] Branscomb, A. Beyond deregulation: Designing the information infrastructure. *The Information Society Journal 1*, 3 (1982), 167–190.

[36] Brinkhoff, T. The impacts of map-oriented Internet applications on Internet clients, map servers and spatial database systems. In *9th International Symposium on Spatial Data Handling* (Beijing, China, August, 10-12 2000).

[37] Brown, A., Johnston, S., and Kelly, K. Using service-oriented architecture and component-based development to build web service applications. Tech.

rep., Rational Software Corporation, 2002. Retrieved December 16, 2008 from http://download.boulder.ibm.com/ibmdl/pub/software/dw/rational/pdf/2169.pdf.

[38] Brown, P. F., and Hamilton, R. M. B. A. Reference model for service oriented architecture 1.0. Tech. rep., 2006. Retrieved December 16, 2008 from http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf.

[39] Brox, C., Bishr, Y., Senkler, K., Zens, K., and Kuhn, W. Toward a geospatial data infrastructure for Northrhine-Westphalia. *Computers, Environment and Urban Systems 26* (2002), 19–37.

[40] Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., and Stal, M. *Pattern-Oriented Software Architecture*. John Wiley and Sons, 1996.

[41] Butterfield, M., Pearlman, J., and Vickroy, S. A system-of-systems engineering geoss: Architectural approach. *IEEE Systems Journal 2*, 3 (2008), 321–332.

[42] Centro de soporte IDEC, Instituto Cartográfico de Cataluña. Infraestructura de Datos Espaciales de Cataluña. http://www.geoportal-idec.net, 2008. Last Accessed on December 16, 2008.

[43] Chang, Y.-S., and Park, H.-D. XML Web Service-based development model for Internet GIS applications. *International Journal of Geographical Information Science 20*, 4 (April 2006), 371–399.

[44] Cipriano, P., and Garretti, L. SITAD: building a local Spatial Data Infrastructure in Piedmont (Italy) within INSPIRE perspective. In *10th EC-GI & GIS Workshop* (Warsaw, Poland, June 23-25 2004), Institute for Environment and Sustainability (IES), Joint Research Centre (JRC), p. 9.

[45] Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, J., Little, R., Nord, R., and Stafford, J. *Documenting Software Architectures: Views and Beyond*. SEI Series in Software Engineering. Addison-Wesley, 2003.

[46] Coleman, D. J., and McLaughlin, J. Defining Global Geospatial Data Infrastructure (GGDI): Components, Stakeholders and Interfaces. In *2nd Global Spatial Data Infrastructure Conference* (Chapel Hill, USA, October 1997).

[47] Comella, S., López, R., Béjar, R., Zarazaga, J., and Muro-Medrano, P. R. Making profit of software interoperabilty in a web enabled water point inventory distributed system. In *Proc. of the First International Workshop on Telegeoprocessing, TELEGEO'99* (Lyon, France, 6-7 May 1999), pp. 71–75.

[48] Cox, S., Cuthbert, A., Lake, R., and Martell, R., Eds. *Geography Markup Language (GML) 2.0*. No. OGC 01-029 in OGC Recommendation Paper. Open GIS Consortium Inc., February 2001.

[49] Craig, W. J. White knights of spatial data infrastructure: The role and motivation of key individuals. *URISA Journal 16*, 2 (2005), 5–13.

[50] Crompvoets, J., Bregt, A., Rajabifard, A., and Williamson, I. Assessing the worldwide developments of national spatial data clearinghouses. *International Journal of Geographical Information Science 18*, 7 (2004), 665–689.

[51] DeLaurentis, D., and Callaway, R. A systems-of-systems perspective for public policy decisions. *Review of Policy Research 21*, 6 (2004), 829–837.

[52] Edmondson, S., Christensen, J., Marks, J., and Shieber, S. A general cartographic labeling algorithm. *Cartographica 33*, 4 (1997), 13–23.

[53] Erl, T. *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services*. Prentice Hall, Indianapolis, Indiana, 2004.

[54] Estrada-Peña, A. Geostatistics and remote sensing as predictive tools of tick distribution: a cokriging system to estimate ixodes scapularis (acari: Ixodidae) habitat suitability in the United States and Canada from advanced very high resolution radiometer satellite imagery. *Journal of Medical Entomology 35* (1998), 989–995.

[55] European Commission. Report on Europe and the Global Information Society: Recommendations of the High-level Group on the Information Society to the Corfu European Council. EU Commission - COM Document Supplement No. 2/94, European Union Commission. Bangemann Report., 1994.

[56] European Commission. INSPIRE State of Play Reports. http://inspire.jrc.ec.europa.eu/state_of_play.cfm, 2008. Last Accessed on December 16, 2008.

[57] European Parliament and The European Council. Directive of the European Parliament and of the Council establishing an Infrastructure for Spatial Information in the European Community (INSPIRE). Joint text approved by the Conciliation Committee, provided for in Article 251(4) of the EC Treaty. 2004/0175(COD), PE-CONS 3685/06, January 2007. Retrieved January 31, 2007 from http://register.consilium.europa.eu/pdf/en/06/st03/st03685.en06.pdf.

[58] Evans, J. D., Ed. *A Geospatial Interoperability Reference Model*. No. 1.1. FGDC Geospatial Applications and Interoperability Working Group, December 2003. Retrieved January 31, 2007 from http://gai.fgdc.gov/girm/v1.1/.

[59] Evans, J. D., Ed. *Web Coverage Service (WCS), Version 1.0.0.* No. OGC 03-065r6. Open GIS Consortium Inc., August 2003.

[60] Federal Geographic Data Committee. National Spatial Data Infrastructure. http://www.fgdc.gov/nsdi/nsdi.html, 2008. Last Accessed on December 16, 2008.

[61] Federal Geographic Data Committee (FGDC). Content Standard for Digital Geospatial Metadata, version 2.0. Document FGDC-STD-001-1998, Metadata Ad Hoc Working Group, 1998.

[62] Fernández, P., Béjar, R., Latre, M. Á., Valiño, J., Bañares, J. A., and Muro-Medrano, P. R. Web mapping interoperability in practice, a Java approach guided by the OpenGis web map server interface specification. In *Proc. of the 6th European Commission GI&GIS Workshop, The Spatial Information Society - Shaping the Future* (Lyon, France, 28-30 June 2000).

[63] Fisher, D. A. An emergent perspective on interoperation in systems of systems. integration of software-intensive systems. Technical Report CMU/SEI-2006-TR-003, Carnegie Mellon University. Software Engineering Institute, 2006.

[64] Fisher, D. A., Meyers, B. C., and Place, P. Conditions for achieving network-centric operations in systems of systems. Technical note CMU/SEI-2007-TN-003, Carnegie Mellon University. Software Engineering Institute, 2007. Retrieved October 27, 2008 from http://www.sei.cmu.edu/pub/documents/07.reports/07tn003.pdf.

[65] Fonseca Nobre, F., and Sá Carvalho, M. *GIS for Health and the Environment.* IDRC, 1995, ch. Spatial and Temporal Analysis of Epidemiological Data.

[66] Fowler, M. *Analysis Patterns*. Addison Wesley Longman Inc., 1997.

[67] Fritz, M., and Beusen, P. Spatial Data Infrastructures in Spain: State of play 2007. Tech. rep., 2008. Retrieved December 15, 2008 from http://inspire.jrc.ec.europa.eu/reports/stateofplay2007/rcr07ESv102.pdf.

[68] Gallego, F. J., Kay, J., Léo, O., Delincé, J., Fine, J., Ambrosio, L., Dellaportas, P., and Balli, F. Estimating the number of olive trees in the European Union. In *52nd Session of the International Statistical Institute* (1999).

[69] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. *Design Patterns. Elements of Reusable Object-Oriented Software*. Addison Wesley, 1995.

[70] GeoConnections. The Canadian Geospatial Data Infrastructure Architecture Description Version 2.0. Tech. rep., 2005.

[71] GeoConnections. GeoConnections - Mapping the future together online. http://cgdi.gc.ca/en/index.html, 2008. Last accessed on December 16, 2008.

[72] Georgiadou, Y. SDI ontology and implications for research in the developing world. *International Journal of Spatial Data Infrastructures Research 1* (2006), 51–64. Retrieved December 15, 2006 from http://ijsdir.jrc.it/editorials/georgiadou.pdf.

[73] Georgiadou, Y., and Groot, R. Capacity building aspects for a geospatial data infrastructure (gdi). In *5th Global Spatial Data Infrastructure Conference* (Cartagena de Indias, Colombia, May 2001).

[74] Georgiadou, Y., Puri, S. K., and Sahay, S. Towards a potential research agenda to guide the implementation of spatial data infrastructures - a case study

from India. *International Journal of Geographical Information Science 19*, 10 (2005), 1113–1130.

[75] Goodchild, M. F. The alexandria digital library project: Review, assesment, and prospects. *D-Lib Magazine 10*, 5 (2004). Retrieved December 15, 2006 from http://www.dlib.org/dlib/may04/goodchild/05goodchild.html.

[76] Gorod, A., Gove, R., Sauser, B., and Boardman, J. System of systems management: A network management approach. In *IEEE International Conference on System of Systems Engineering, 2007. SoSE '07.* (2007), pp. 1–5.

[77] Grus, L., Crompvoets, J., and Bregt, A. K. Multi-view SDI assessment framework. *International Journal of Spatial Data Infrastructures Research 2* (2007), 33–53.

[78] GSDI Technical Working Group and contributors. *Developing Spatial Data Infrastructures: The SDI Cookbook v.2.0.* Global Spatial Data Infrastructure (http://www.gsdi.org), January 2004.

[79] Hanseth, O. Social shaping of information infrastructure: On being specific about the technology. In *Information Technology and Changes in Organizational Work: Proceedings of the IFIP WG8. 2 Working Conference on Information Technology and Changes in Organizational Work, December 1995* (1996), Springer.

[80] Hanseth, O., and Lundberg, N. Designing work oriented infrastructures. *Computer Supported Cooperative Work 10*, 3-4 (2001), 347–372.

[81] Hanseth, O., and Monteiro, E. *Understanding Information Infrastructure.* (forthcoming book), manuscript available on http://heim.ifi.uio.no/ oleha/Publications/bok.html, 1998.

[82] Harrie, L., Zhang, Q., and Ringberg, P. A case study of combined text and icon placement. In *International Cartographic Conference 2005: Mapping Approaches into a Changing World* (A Coruña, Spain, July 2005).

[83] Harrison, J. Ogc web services. geoprocessing and the new web computing paradigm. *Geoinformatics*, 5 (2002), 18–21.

[84] Heath, B. P., McArthur, D. J., McClelland, M. K., and Vetter, R. J. Metadata lessons from the ilumina digital library. *Communications of the ACM 48*, 7 (2005), 68–74.

[85] Hipel, K., Jamshidi, M., Tien, J., and White, C. The future of systems, man, and cybernetics: Application domains and research methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 37*, 5 (2007), 726–743.

[86] Hjelmager, J., Delgado, T., Moellering, H., Cooper, A., Danko, D., Huet, M., Aalders, H., and Martynenko, A. Developing a modeling for the spatial data infrastructure. In *XXII International Cartographic Conference* (A Coruña, Spain, July 2005), The International Cartographic Association (ICA-ACI).

[87] Hofmeister, C., Nord, R., and Soni, D. *Applied Software Architecture.* Addison-Wesley, 2000.

[88] IEEE Architecture Working Group. IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. IEEE Standard IEEE Std 1471-2000, IEEE Standards Association, 2000.

[89] Instituto Geográfico Nacional como Secretaría Técnica del Consejo Superior Geográfico. Infraestructura de Datos Espaciales de España. http://www.idee.es/, 2008. Last Accessed on December 16, 2008.

[90] International Organization for Standardization (ISO). Geographic information - Metadata. ISO 19115:2003, 2003.

[91] ISF Working Group. Implementing structures and funding position paper. Tech. rep., INSPIRE Working Groups, 2002.

[92] ISO. *Geographic Information - Services*. International standard; ISO 19119. International Organization for Standardization, Genève, Switzerland, 2003.

[93] ISO / IEC. *Information technology - Open Distributed Processing - Reference Model: Architecture*, first edition ed. International standard; ISO/IEC 10746-3. International Organization for Standardization / International Electrotechnical Commission, Geneva, Switzerland, 1996.

[94] ISO / IEC. *Information technology - Open Distributed Processing - Reference Model: Foundations*, first edition ed. International standard; ISO/IEC 10746-2. International Organization for Standardization / International Electrotechnical Commission, Geneva, Switzerland, 1996.

[95] ISO / IEC. *Information technology - Open Distributed Processing - Reference Model: Architectural Semantics*, first edition ed. International standard;

ISO/IEC 10746-4. International Organization for Standardization / International Electrotechnical Commission, Geneva, Switzerland, 1998.

[96] ISO / IEC. *Information technology - Open Distributed Processing - Reference Model: Overview*, first edition ed. International standard; ISO/IEC 10746-1. International Organization for Standardization / International Electrotechnical Commission, Geneva, Switzerland, 1998.

[97] ISO / IEC. *Information technology - Open distributed processing - Reference model - Enterprise language*, second edition ed. International standard; ISO/IEC 15414:2006. International Organization for Standardization / International Electrotechnical Commission, Geneva, Switzerland, 2006.

[98] ISO / IEC. *Information technology — Open distributed processing — Use of UML for ODP system specifications.* Final Draft International Standard; ISO/IEC 19793:2007. International Organization for Standardization / International Electrotechnical Commission, Montréal, Québec, Canada, 2007.

[99] Janée, G., Frew, J., and Hill, L. L. Issues in georeferenced digital libraries. *D-Lib Magazine 10*, 5 (2004). Retrieved December 15, 2006 from http://www.dlib.org/dlib/may04/janee/05janee.html.

[100] Jeff de la Beaujardiere, Ed. *OGC Web Map Service Interface 1.3.0.* No. OGC 03-109r1. Open GIS Consortium Inc., January 2004.

[101] Jens Fitzke, Klaus Greve, M. M. A. P. Building sdis with free software – the degree project. In *GSDI 7 - Spatial Data Infrastructures for a Sustainable Future* (2004).

[102] Kim, D.-H., Kim, K.-S., Choi, H., and Lee, J.-H. The design and implementation of open gis service component. In *IEEE International Geoscience and Remote Sensing Symposium.* (2001), vol. 4, pp. 1922–1924.

[103] Kruchten, P., Obbink, H., and Stafford, J. The past, present and future of software architecture. *IEEE Software 23*, 2 (2006), 22–30.

[104] Kuhn, W. Geospatial semantics: Why, of what and how? In *Journal on Data Semantics III*, S. Spaccapietra, Ed., vol. 3534 of *Lecture Notes in Computer Science, LNCS*. Springer, 2005, pp. 1–24.

[105] Lacasta, J., Nogueras-Iso, J., Béjar, R., Muro-Medrano, P. R., and Zarazaga-Soria, F. J. A Web Ontology Service to facilitate interoperability within a Spatial Data Infrastructure: applicability to discovery. *Data & Knowledge Engineering 63*, 3 (2007), 947–971.

[106] Lacasta, J., Nogueras-Iso, J., López-Pellicer, F. J., Muro-Medrano, P. R., and Zarazaga-Soria, F. J. ThManager: An open source tool for creating and visualizing SKOS. *Information Technology and Libraries 26*, 3 (2007), 39–51.

[107] Lalonde, W., Ed. *Styled Layer Descriptor Implementation Specification.* No. OGC 02-070 in Proposed OpenGIS OGC Implementation Specification. Open GIS Consortium Inc., September 2002.

[108] Latre, M., Béjar, R., Fernández, P., Álvarez, P., and Muro-Medrano, P. Trying Java technology in a geologic-mining information system distributed over an inter/intranet environment. In *Proc. of the Second International Symposium on Telegeoprocessing, TELEGEO'00* (Sophia-Antipolis, France, 10-12 May 2000), pp. 93–101.

[109] Latre, M. Á., Zarazaga-Soria, F. J., Nogueras-Iso, J., Béjar, R., and Muro-Medrano, P. SDIGER: A cross-border inter-administration SDI to support WFD information access for Adour-Garonne and Ebro river basins. In *Proceedings of the 11th EC GI & GIS Workshop, ESDI Setting the Framework* (Alguero, Italy, June 2005).

[110] Lautenbacher, C. C. The global earth observation system of systems: Science serving society. *Space Policy 22*, 1 (2006), 8–11.

[111] Lee, S., and Gandhi, R. Ontology-based active requirements engineering framework. In *12th Asia-Pacific Software Engineering Conference, 2005. APSEC '05.* (2005), p. 8 pp.

[112] Lemmens, R., Wytzisk, A., de By, R., Granell, C., Gould, M., and van Oosterom, P. Integrating semantic and syntactic descriptions to chain geographic services. *IEEE Internet Computing 10*, 5 (2006), 42–52.

[113] Longley, P., Goodchild, M. F., Maguire, D., and (eds.), D. R. *Geographical Information Systems: Principles, Techniques, Applications and Management. Volume 1*, 2 ed. Wiley, 1999.

[114] Longley, P. A., Goodchild, M. F., Maguire, D. J., and Rhind, D. W. *Geographic Information Systems and Science.* Wiley, 2001.

[115] López Agostini, D., and Vallès Montoliu, M. Intelligent maps through Internet: application to the diffusion of environmental information. In *Telegeo'2000, Second International Symposium on Telegeoprocessing* (Sophia-Antipolis, France, May 2000), pp. 143–152.

[116] Lu, X. An investigation on service-oriented architecture for constructing distributed web gis application. In *IEEE International Conference on Services Computing* (2005), vol. 1, pp. 191–197.

[117] Lupp, M., Ed. *Styled Layer Descriptor profile of the Web Map Service Implementation Specification.* No. OGC 05-078r4 in OGC Implementation Specification. Open Geospatial Consortium Inc., June 2007.

[118] Ma, S., Li, M., and Du, W. Service composition for gis. In *IEEE Congress on Services - Part I* (2008), pp. 168–175.

[119] Maier, M. W. Architecting principles for systems-of-systems. In *6th Annual International Symposium of INCOSE* (Boston, MA, USA, July 1996), pp. 567—-574. Retrieved October 27, 2008 from http://www.infoed.com/Open/PAPERS/systems.htm.

[120] Mapping Science Committee, N. R. C. *Toward a Coordinated Spatial Data Infrastructure for the Nation.* National Academy Press, 1993.

[121] Markov, N. G., and Napryushkin, A. A. Use of remote sensing data as thematic mapping in GIS. In *3rd AGILE Conference on Geographic Information Science* (Helsinki / Espoo, Finland, May, 25-27 2000), pp. 51–55.

[122] Masó, J., and Pons, X. Descargas y animaciones en el navegador de mapas OGC de MiraMon. In *Jornadas de la IDE de España* (Zaragoza, Spain, November 2004).

[123] Masser, I. What's special about SDI related research? *International Journal of Spatial Data Infrastructures Research 1* (2006), 14–23. Retrieved July 23, 2008 from http://ijsdir.jrc.it/index.php/ijsdir/article/viewFile/15/9.

[124] Mayer, W. H. Agriculture - Environment - Health - Integration. In *6th EC-GI&GIS Workshop* (Lyon, France, June 2000). CD-ROM.

[125] Medvidovic, N., and Taylor, R. N. A classification and comparison framework for software architecture description languages. *IEEE Transactions on Software Engineering 26*, 1 (2000), 70–93.

[126] Miller, J., and Mukerji, J., Eds. *MDA Guide Version 1.0.1*. No. omg/2003-06-01. Object Management Group, Inc., June 2003.

[127] Morris, E., Place, P., and Smith, D. System-of-systems governance: New patterns of thought. Technical note CMU/SEI-2006-TN-036, Carnegie Mellon University. Software Engineering Institute, 2006. Retrieved October 27, 2008 from http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06tn036.pdf.

[128] Más Mayoral, S. IDEE: The state of play of setting up the National SDI of Spain. In *9th EC-GI & GIS Workshop. ESDI: Serving the User* (A Coruña, Spain, June 2003).

[129] Nebert, D., Whiteside, A., and Vretanos, P. P., Eds. *OpenGIS Catalogue Services Specification*. No. OGC 07-006r1 in OpenGIS Implementation Specification. Open Geospatial Consortium Inc., February 2007.

[130] Nebert, D. D. Application profile for geospatial metadata or ”GEO”. Application Profile Version 2.2, U.S. Federal Geographic Data Committee, U.S. Geological Survey, 1999. Retrieved August 1, 2008 from http://www.fgdc.gov/standards/projects/GeoProfile.

[131] Nedovic-Budic, Z., and Budhathoki, N. R. Technological and institutional interdependences and SDI - The Bermuda Square? *International Journal of*

Spatial Data Infrastructures Research 1 (2006), 36–50. Retrieved December 15, 2006 from http://ijsdir.jrc.it/editorials/budic_nama.pdf.

[132] Nogueras-Iso, J., Béjar, R., Latre, M. Á., Fernández-Villoslada, E., Muro-Medrano, P. R., and Zarazaga-Soria, F. J. *The European Information Society - Leading the Way with Geo-information.* Lecture Notes in Geoinformation and Cartography. Springer, 2007, ch. Estimating the costs of an SDI-based project, pp. 201–215.

[133] Nogueras-Iso, J., Zarazaga-Soria, F. J., Béjar, R., Álvarez, P. J., and Muro-Medrano, P. R. OGC catalog services: a key element for the development of spatial data infrastructures. *Computers and Geosciences 31*, 2 (2005), 199–209.

[134] Nogueras-Iso, J., Zarazaga-Soria, F. J., Lacasta, J., Béjar, R., and Muro-Medrano, P. R. Metadata standard interoperability: Application in the geographic information domain. *Computers, Environment and Urban Systems 28*, 6 (November 2004), 611–634.

[135] Nogueras-Iso, J., Zarazaga-Soria, F. J., and Muro-Medrano, P. R. *Geographic Information Metadata for Spatial Data Infrastructures - Resources, Interoperability and Information Retrieval.* Springer Verlag, 2005.

[136] O'Reilly, T. What is web 2.0: Design patterns and business models for the next generation of software. http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html?page=1, 2005.

[137] Orfali, R., and Harkey, D. *Client/server programming with Java and CORBA (2nd ed.).* John Wiley & Sons, Inc., New York, NY, USA, 1998.

[138] Panel on Distributed Geolibraries, National Research Council. *Distributed Geolibraries: Spatial Information Resources, Summary of a Workshop.* National Academy Press, 1999. Retrieved December 15, 2006 from http://www.nap.edu/html/geolibraries/.

[139] Peedell, S., Kay, S., and Giardino, G. Seeing the (olive) trees from the wood - using GIS in Europe for olive tree identification. In *ESRI User Conference* (San Diego, USA, July 1998).

[140] Peng, Z.-R., and Tsou, M.-H. *Internet GIS: Distributed Geographic Information Services for the Internet and Wireless Networks.* Wiley, 2003.

[141] Percivall, G., Ed. *OGC Reference Model.* No. OGC 08-062r4. Open Geospatial Consortium Inc., November 2008.

[142] Pereira, P., Neves, N., Silva, J. P., Muchaxo, J., and Câmara, A. *Interoperating Geographic Information Systems.* Kluwer Academic Publishers, 1999, ch. Interoperability of geographic information: from the spreadsheet to virtual environments.

[143] Portolés-Rodríguez, D., Álvarez, P., Béjar, R., and Muro-Medrano, P. Idezar: an example of user needs, technological aspects and the institutional framework of a local sdi. In *Proc. of the 11th EC GI & GIS Workshop, ESDI Setting the Framework* (Alghero (Sardinia), Italia, June 2005), pp. 56–58.

[144] Putman, J. R. *Architecting with RM-ODP.* Prentice Hall, Upper Saddle River, New Jersey, 2000.

[145] Putz, S. Interactive information services using World Wide Web Hypertext. In *First International Conference on the World-Wide Web* (Geneva, Switzerland, 1994).

[146] Rajabifard, A., Binns, A., Masser, I., and Williamson, I. The role of sub-national government and the private sector in future spatial data infrastructures. *International Journal of Geographical Information Science 20*, 7 (2006), 727–741.

[147] Rajabifard, A., and Williamson, I. P. Spatial data infrastructures: Concept, SDI hierarchy and future directions. In *Geomatics'80 Conference* (Tehran, Iran, April-May 2001).

[148] Rajabifard, A., Williamson, I. P., Holland, P., and Johnstone, G. From local to global SDI initiatives: a pyramid building blocks. In *4th GSDI Conference* (Cape Town, South Africa, August 2000).

[149] Rasmussen, J. E., Rasmussen, L. E., and Ma, S. Generating and serving tiles in a digital mapping system. Patent. U.S. Classification: 345552000 ; International Classification: G06T011/40.

[150] Robinson, A. H., Morrison, J. L., Muehrcke, P. C., Kimerling, A. J., and Guptill, S. C. *Elements of Cartography.* Wiley, 1995.

[151] Rumbaugh, J., Jacobson, I., and Booch, G. *The Unified Modeling Language Reference Manual (2nd Edition).* Addison-Wesley Professional, 2004.

[152] Sage, A. P., and Cuppan, C. D. On the systems engineering and management of systems of systems and federations of systems. *Information Knowledge Systems Management 2*, 4 (December 2001), 325–345.

[153] Sancho-Jiménez, G., Béjar, R., Latre, M. A., and Muro-Medrano, P. R. A method to derivate SOAP interfaces and WSDL metadata from the OGC web processing service mandatory interfaces. In *Advances in Conceptual Modeling-Challenges and Opportunities* (Barcelona, Spain, 2008), p. 375.

[154] Sayar, A., Pierce, M., and Fox, G. Developing gis visualization web services for geophysical applications. In *ISPRS 2005 Spatial Data Mining Workshop* (Ankara, Turkey).

[155] Schäffer, B., Ed. *OWS 5 SOAP/WSDL Common Engineering Report*. No. OGC 08-009r108-009r1 in OGC Discussion Paper. Open Geospatial Consortium Inc., January 2008.

[156] Star, S. L., and Ruhleder, K. Steps toward an ecology of infrastructure: Design and access for large information spaces. *Information Systems Research 7*, 1 (1996), 111–134.

[157] Strijk, T. *Geometric Algorithms for Cartographic Label Placement*. PhD thesis, Department of Computer Science, Utrecht University, January 2001.

[158] Sun Microsystems, I. Java SE Desktop Technologies - Java Beans. http://java.sun.com/javase/technologies/desktop/javabeans/index.jsp. Retrieved December 4, 2008.

[159] Sun Microsystems, I. Java Servlet Technology. http://java.sun.com/products/servlet/index.jsp. Retrieved December 4, 2008.

[160] Sun Microsystems, I. The Source for Java Developers. http://java.sun.com. Retrieved December 4, 2008.

[161] The Global Spatial Data Infrastructure Association. GSDI Association. http://www.gsdi.org/, 2008. Last accessed on December 16, 2008.

[162] Thompson, B., Warnest, M., and Chipchase, C. *Developing Spatial Data Infrastructures: From Concept to Reality.* Taylor and Francis, London, New York, 2003, ch. State SDI Development: A Victorian Perspective, pp. 147–164.

[163] Tolosana-Calasanz, R., Nogueras-Iso, J., Béjar, R., Muro-Medrano, P. R., and Zarazaga-Soria, J. Semantic interoperability based on Dublin Core hierarchical one-to-one mappings. *Int. J. Metadata, Semantics and Ontologies 1*, 3 (2006), 183–188.

[164] Tuffery, C., and Durand, H. A web site for diffusing and on-line up-dating by users of georeferenced raster maps of land-cover of wetlands an their basin-slopes in region Languedoc-Roussillon. In *Telegeo'2000, Second International Symposium on Telegeoprocessing* (Sophia-Antipolis, France, May 2000), pp. 75–80.

[165] United Nations Geographic Information Working Group. United Nations Spatial Data Infrastructure - UNSDI. http://www.ungiwg.org/unsdi.htm, 2008. Last accessed on December 16, 2008.

[166] U.S. Congress. High Performance Computing and Communications Act of 1991 (HPCA, Pub.L. 102-194, enacted 1991-12-09). Superintendent of Documents, Congressional Sales Office, U.S. Government Printing Office, Washington DC 20402, 1991.

[167] Valerdi, R., Axelband, E., Baehren, T., Boehm, B., Dorenbos, D., Jackson, S., Madni, A., Nadler, G., Robitaille, P., and Settles, S.

A research agenda for systems of systems architecting. *International Journal of System of Systems Engineering 1*, 1/2 (2008), 171–188.

[168] van Dijk, S. *Genetic Algorithms for Map Labeling*. PhD thesis, Department of Computer Science, Utrecht University, November 2001.

[169] van Dijk, S., van Kreveld, M., Strijk, T., and Wolff, A. Towards an evaluation of quality for names placement methods. *International Journal of Geographical Information Science 16*, 7 (2002), 641–661.

[170] van Kreveld, M., Étienne Schramm, and Wolff, A. Algorithms for the placement of diagrams on maps. In *12th International Symposium ACM GIS* (November 2004), D. Pfoder, I. F. Cruz, and M. Ronthaler, Eds., pp. 222–231.

[171] Vretanos, P. A., Ed. *Web Feature Service Implementation Specification 1.0.0*. No. OGC 02-058 in OpenGIS Implementation Specification. Open GIS Consortium Inc., May 2002.

[172] Warnest, M. *A Collaboration Model for National Spatial Data Infrastructure in Federated Countries*. PhD thesis, Department of Geomatics, The University of Melbourne, February 2005. Retrieved December 15, 2006 from http://eprints.infodiv.unimelb.edu.au/archive/00001361/.

[173] Webster, M. D., Forest, C. E., Reilly, J. M., Sokolov, A. P., Stone, P. H., Jacoby, H. D., and Prinn, R. G. Uncertainty analysis of global climate change projections. Joint Program on the Science and Policy of Global Change Report 73, Massachusetts Institute of Technology, 2001. Retrieved August 5, 2008 from http://web.mit.edu/globalchange/www/rpt73.html.

[174] Whiteside, A., Ed. *OpenGIS web services architecture description.* No. OGC 05-042r1 in OpenGIS Discussion Paper. Open Geospatial Consortium Inc., October 2005.

[175] Wickramasinghe, N., Chalasani, S., Boppana, R., and Madni, A. Healthcare system of systems. In *IEEE International Conference on System of Systems Engineering, 2007. SoSE '07.* (2007), pp. 1–6.

[176] World Wide Web Consortium (W3C). Web Services Activity. http://www.w3.org/2002/ws/, 2008. Last accessed on December 16, 2008.

[177] Wytzisk, A., and Sliwinski, A. Quo vadis SDI? In *7th AGILE Conference on Geographic Information Science* (Heraklion, Greece, May 2004), pp. 43–49.

[178] Yasnoff, W. A., Humphreys, B. L., Overhage, J. M., Detmer, D. E., Brennan, P. F., Morris, R. W., Middleton, B., Bates, D. W., and Fanning, J. P. A consensus action agenda for achieving the national health information infrastructure. *Journal of the American Medical Informatics Association 11*, 4 (2004), 332–338.

[179] Yétongnon, K., Benslimane, D., Laurini, R., Leclercq, E., Jouanot, F., and Cullot, N. Gis interoperability: an overview. In *Telegeo 2000, Second Internatinal Symposium on Telegeoprocessing* (Sophia-Antipolis, France, May 2000), pp. 185–204.

[180] Zarazaga-Soria, F., Nogueras-Iso, J., Béjar, R., and Muro-Medrano, P. Political aspects of spatial data infrastructures. In *Electronic Government*, R. Traunmüller, Ed., vol. 3183 of *Lecture Notes in Computer Science, LNCS*. Springer, 2004, pp. 392–395.

[181] Zarazaga-Soria, F. J., Lacasta, J., Nogueras-Iso, J., Torres, M. P., and Muro-Medrano, P. R. A Java tool for creating ISO/FGDC geographic metadata. In *Geodaten- und Geodienste-Infrastrukturen - von der Forschung zur praktischen Anwendung. Beiträge zu den Münsteraner GI-Tagen 26./27. Juni 2003* (Münster, Germany, 26-27 June 2003), vol. 18, IFGIprints, pp. 17–30.

[182] Zarazaga-Soria, F. J., Muro-Medrano, P. R., Béjar, R., Latre, M. Á., and Barrera Francés, J. Aero 1.0. Registro General de la Propiedad Intelectual, Gobierno de Aragón, 2006. Programa de ordenador, asiento registral 10/2006/112.

[183] Zarazaga-Soria, F. J., Muro-Medrano, P. R., Béjar, R., Latre, M. Á., and Laiglesia Martínez, S. SPIGA 4.2. Registro General de la Propiedad Intelectual, Gobierno de Aragón, 2006. Programa de ordenador, asiento registral 10/2006/92.

[184] Zarazaga-Soria, F. J., Muro-Medrano, P. R., Álvarez, P. J., Béjar, R., and Gayán Asensio, D. JGISView 5.1. Registro General de la Propiedad Intelectual, Gobierno de Aragón, 2006. Programa de ordenador, asiento registral 10/2006/108.

[185] Zarazaga-Soria, F. J., Muro-Medrano, P. R., Álvarez, P. J., Béjar, R., and Rioja Guedán, R. Geoportal 1.0.2. Registro General de la Propiedad Intelectual, Gobierno de Aragón, 2006. Programa de ordenador, asiento registral 10/2006/118.

[186] Zarazaga-Soria, F. J., Muro-Medrano, P. R., Nogueras-Iso, J., Béjar, R., and Lacasta, J. WOS 1.0. Registro General de la Propiedad Intelectual, Gobierno de Aragón, 2006. Programa de ordenador, asiento registral 10/2006/94.

[187] Zarazaga-Soria, F. J., Muro-Medrano, P. R., Nogueras-Iso, J., Álvarez, P. J., Béjar, R., and Laiglesia Martínez, S. JGISView Web 2.0. Registro General de la Propiedad Intelectual, Gobierno de Aragón, 2006. Programa de ordenador, asiento registral 10/2006/109.

[188] Zarazaga-Soria, F. J., Muro-Medrano, P. R., Nogueras-Iso, J., Álvarez, P. J., Béjar, R., and Laiglesia Martínez, S. JMapServer 2.1. Registro General de la Propiedad Intelectual, Gobierno de Aragón, 2006. Programa de ordenador, asiento registral 10/2006/110.

[189] Zarazaga-Soria, F. J., Nogueras-Iso, J., Latre, M. Á., Rodríguez, A., López, E., Vivas, P., and Muro-Medrano, P. *Research and Theory in Advancing Spatial Data Infrastructure Concepts*. ESRI Press, 2007, ch. Providing SDI Services in a Cross-Border Scenario: the SDIGER Project Use Case, pp. 107–119.

[190] Zhang, Q., and Harrie, L. Real time map labeling for personal navigation. In *12th Conference on Geoinformatics* (Gävle, Sweden, June 2004), pp. 39–46.

[191] Zhang, S., and Goddard, S. OpenGIS Conforming Map-Feature Server Implementation Specifications in Component-based Distributed Systems. *Environmental Informatics Archives 1* (2003), 487–496.