(Draft) International Journal of Geographical Information Science. 2009, vol. 23, nº 3, p. 271-294.

International Journal of Geographical Information Science
Vol. XX, No. XX, XXXXX XXXX, 1–19

An architectural style for Spatial Data Infrastructures

R. BÉJAR*, M. Á. LATRE, J. NOGUERAS-ISO, P. R. MURO-MEDRANO and F. J. ZARAZAGA-SORIA

Department of Computing and Systems Engineering, University of Zaragoza, c/ María de Luna 1, 50018, Zaragoza, Spain

(*v1.0 released June 2007 - THIS IS A PRE-PRINT VERSION OF THE PAPER, SUBMITTED TO THE INTERNATIONAL JOURNAL OF GEOGRAPHICAL INFORMATION SCIENCE*)

This work proposes an architectural style, a pattern, for SDIs. This style provides a tool and a shared vocabulary to help system architects to design these infrastructures, and facilitates the exchange of knowledge about them. This style is defined under the component-and-connector architectural viewtype, extending the client-server and shared-data styles. The style has been created after analyzing six of the most relevant SDI and geoservices architectural proposals. Several architectural elements that these proposals have not properly addressed are considered. Three real projects, with published architectural views or models, have been examined to verify the applicability of the style. The proposed style offers a systematization and refinement of knowledge about SDIs, grounded in well-known concepts in software architecture.

Keywords: SDI; Software Architecture; Architectural Style; Pattern;

1 Introduction

From different points of view, SDIs have some characteristics of digital libraries (Panel on Distributed Geolibraries, National Research Council (1999)) and of information infrastructures (IIs) (Georgiadou *et al.* (2005)). From the II point of view, an SDI is formed by several interconnected SDIs; but when one of these SDIs is responsibility of one organization, it is normally 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 is necessary to address its design, creation, and maintenance and is the main focus of this paper.

The original definitions of SDIs already include, directly or indirectly, the necessity to provide search, visualization and data download services (GSDI Technical Working Group and contributors (2004)). Nevertheless the current research trends in the evolution of distributed interoperable GIS, based in standard (Harrison (2002), Bernard *et al.* (2003)) and semantically enabled (Kuhn (2005), Lemmens *et al.* (2006)), geographic Web services (Chang and Park (2006)), have made some authors argue that SDIs will soon include all kind of geographic Web services (Bernard and Craglia (2005), Wytzisk and Sliwinski (2004)). 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 are the one by Evans (2003), which describes the USA National SDI, the one by GeoConnections (2005) about the Canadian one and the initial proposals for the European SDI established by the ISF Working Group (2002). 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.

*Corresponding author. Email: rbejar@unizar.es

International Journal of Geographical Information Science
ISSN XXXXXX print / ISSN XXXXX online © XXXX Taylor & Francis
http://www.tandf.co.uk/journals
DOI: XXXXX

- 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. Service Oriented Architecture (SOA) (Erl (2004)) 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 work, 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 paper is structured as follows: first some definitions are given about software architecture and the concepts of view, viewtype and style. Then an architectural style for SDI is defined, after analyzing the architectures proposed by six SDI and geographic services reference models. In section 4 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 issues are discussed and conclusions are drawn.

2 Software architecture

According to Kruchten *et al.* (2006), 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 can best be designed and analyzed at the system level. Clements *et al.* (2003) sustain that it is the structure of structures of a system, formed by elements, their visible properties and the relationships among them. The IEEE Architecture Working Group (2000) 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 many others, share the basics: 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.

The software architecture of a system is an inherent property of this system, but expressing or documenting this architecture may be a very complex task. The approach that has been consolidated over the latest 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 and 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.

2.1 *Architectural views*

The 'views and beyond' proposal by Clements *et al.* (2003), or the IEEE Architecture Working Group (2000) recommended practice for architectural description of software-intensive systems, 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, as the 'Siemens four views' described in Hofmeister *et al.* (2000) or a fixed sets of viewpoints as the ISO and ITU-T Reference Model for Open Distributed Processing, which can be consulted in Putman (2000). Architectural styles are refinements of the viewtypes, that 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).

In the rest of this paper the 'views and beyond' proposal is followed, so the style presented here is under the umbrella of one of the viewtypes defined by Clements *et al.* (2003), the Component-and-Connector (C&C) viewtype, which is described in the next subsection. The guidelines in Clements *et al.* (2003) (chapter 6.5) for creating new styles have been taken as a base. This decision does not preclude a more formal approach that could be created later, based on the style presented in this paper, in order to provide a less ambiguous representation of SDI architectures, which would allow for some automatic analysis capabilities and also to help to prevent inconsistencies.

2.2 *The C&C viewtype*

Clements *et al.* (2003) (p. 103) indicate that 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.

3 An SDI style for the C&C viewtype

In this section an SDI architectural style for the C&C viewtype is defined. The objective behind defining this style is to capture, unify and systematize the previous knowledge on SDI architectural models, and to explicitly take into consideration elements that typically are not considered in these models (i.e. constraints), or considered only implicitly (i.e. data stores). So, 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 the architecture of an SDI, considering 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 2, to do this several views, following different viewtypes and styles, would be needed.
- 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 Clements *et al.* (2003) (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 paper is a specialization of a hybrid style which combines these two:

- **Shared-Data**: 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**: Shows asymmetric interaction among components, from clients to servers. 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*.

In the next sections, the elements of a new style for SDI are described. 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 (Béjar *et al.* (2003), Béjar *et al.* (2004), Latre *et al.* (2005), Portolés-Rodríguez *et al.* (2005)) an 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.

3.1 *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/TC 211 standard on geographic information services (ISO 19119, ISO (2003)): 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 Open Geospatial Consortium Web services architecture description (Whiteside (2005)): 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 Geospatial Interoperability Reference Model (Evans (2003)): 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 INSPIRE position paper on architecture (AST Working Group (2002)).
- The Canadian Geospatial Data Infrastructure Architecture Description (GeoConnections (2005)): 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, European Parliament and The European Council (2007)): relevant because it establishes the minimum requisites for all national SDI 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 paper by Lars Bernard, Ioannis Kanellopoulos, Alessandro Annoni and Paul Smits, from the European Commision Joint Resarch Center, presenting the initial steps leading to the establishment of the European Geographic Information Portal (Bernard *et al.* (2005)). 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 (GSDI Technical Working Group and contributors (2004)) 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 *Component types*

The component types in this SDI style are specializations of those in the client-server and shared-data styles defined in Clements *et al.* (2003). 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 assume a Service Oriented Architecture (Erl (2004)), deployed over Internet protocols with XML as the data exchange format, i.e. Web Services (Booth *et al.* (2004)), as the chosen technologic platform; this is also the case of this work. The component types have been chosen because they play relevant roles in SDIs, although not all of them will be present in every SDI. Their names have been selected from the bibliography when there seemed to be a high degree of consensus, and when this has not been possible they have been chosen to try 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 paper 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.

A diagram showing the hierarchical relationships among these component types, and with those in the client-server and shared-data styles, is shown in figure 1. This is a UML class diagram where classes represent component types. The table 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 (Booth *et al.* (2004)).
- **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 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 directly to, spatial data sets, or parts of them.
  - **Catalog Service**: These make it possible to discover, explore and evaluate datasets, services etc., by means of the metadata about them which catalogs publish.
  - **Gazetteer Service**: They offer geocoding functionalities, which link toponyms and their spatial location.
  - **Knowledge Model Service**: 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, being knowledge models, data models, schemas, ontologies, thesauri or any other explicit conceptualization of knowledge in a domain.
- **Dataset Repository**: A repository which holds datasets, being datasets identifiable collections of data.
  - **Spatial Dataset Repository**: A repository which holds spatial datasets, being spatial datasets identifiable collections of spatial data (i.e. data with a direct or indirect reference to a specific location).

(Draft) International Journal of Geographical Information Science. 2009, vol. 23, nº 3, p. 271-294.

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 service)	✗	✗	✗
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 1. SDI style component types compared with those in the other architectures studied

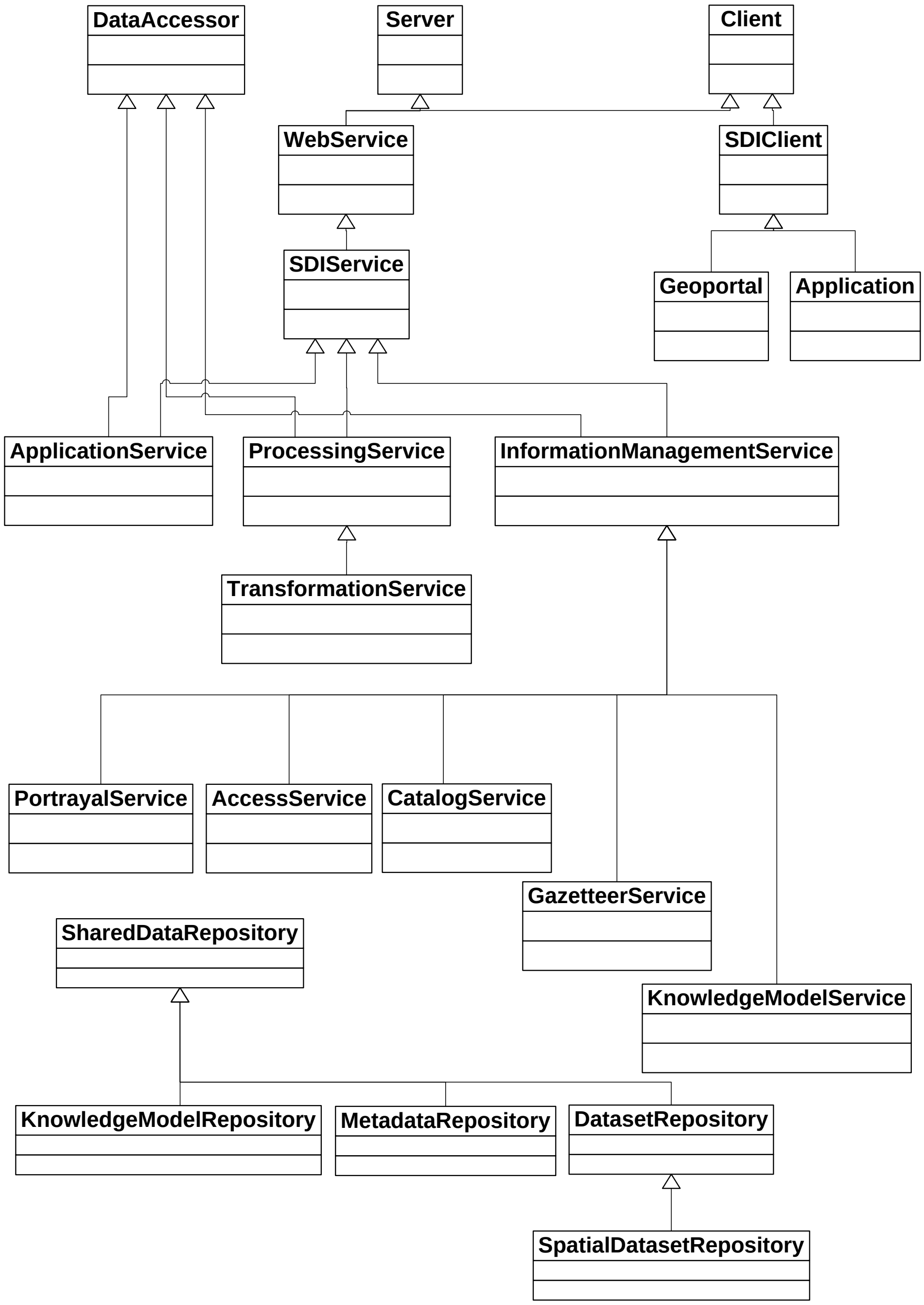

Figure 1. Hierarchical relationships among SDI style component types

3.3 *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 as connectors in general are barely mentioned; it could also be possible that they have not been considered at all. After some study, new relevant types of connectors for the style defined in this work have not been found, not even refinements of those provided by the Client-Server and Shared-Data styles, so 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).

3.4 *Properties*

As with the connector types (see subsection 3.3), it has not been possible to find relevant properties for SDIs which where present in a significant number of architectural proposals, and not in the generic C&C styles (Client-Server and Shared-Data). But among the several properties for these styles suggested by Clements *et al.* (2003), there are some which are used in some studied SDI architectures, so only those are listed here:

- **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 exclusive. A system architect may consider useful adding others when designing her SDI 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.

3.5 *Constraints*

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

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**: In this architecture some ideas which indeed are constraints are described. They would be more clear if they were separated and made explicit. The constraints designed for the style in this paper 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 to 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 access to services and to content repositories (direct data access), and services access to

other services and to 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 to 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 paper.
- **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 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 that extends 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 to clarify some points, and to highlight a few consequences of these constraints:

- Some constraints may seem too restrictive: for example, a Catalog Service is not allowed to read from a dataset repository, and some could argue this could be desirable. This is not a problem: when defining an architectural view following this style, one can define a component 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 the SDI she is designing a catalog component which accesses metadata and datasets is the best solution. This style allows for that while making it explicit that this catalog 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 her design, relating her catalog component to the component types defined for this style. This also makes her 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 3.4), this list of constraints is not exclusive; 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 useful in the experience of the authors with SDI projects.

4 Analysis of real SDI architectures

In this section three different projects are analyzed, in order to determine if real SDIs have an architecture which fits 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 near to the C&C viewtype.

4.1 *Architecture of the Galicia CMA SDI*

Galicia is a Spanish Region, NUTS-2 in the EU terminology, 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 their environmental department, (Consellería de Medio Ambiente, CMA), with water use, disposal of waste and protected natural environments among its other responsibilities. This department had found the same kind of problems with geographic information that SDIs address: 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, in the year 2001, was to develop a geographic information system for this department, but following INSPIRE principles and recommendations in architecture and standards, thus effectively building an SDI. This project should be developed using the available commercial software licenses in the CMA in a COTS (commercial off-the-shelf) approach. This infrastructure was designed to become the core of a future Galician SDI, and is described in some detail in Béjar *et al.* (2003). In this paper, there is an architectural vision of this SDI, referred to as a 'Service Oriented Architecture', that is shown in figure 2.

This architecture is shown in a layered way, focusing on its components and some of their properties: there are neither explicit connectors nor constraints (though some of these can be extracted from the text of the paper). Regarding the components, and following their function as explained in the paper, they all can be matched to some of the component types proposed in the SDI style:

- In the layer 'Data and Metadata Sources':
  - Vectorial 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**.

From the text of the paper some connectors and constraints can be deduced as implied. At least there is one that is quite clear and which also matches the proposed style:

- 'User applications are built *on top of distributed services*' (p. 94, emphasis added). This implies a connector between user applications and distributed services and follows two of the defined constraints:

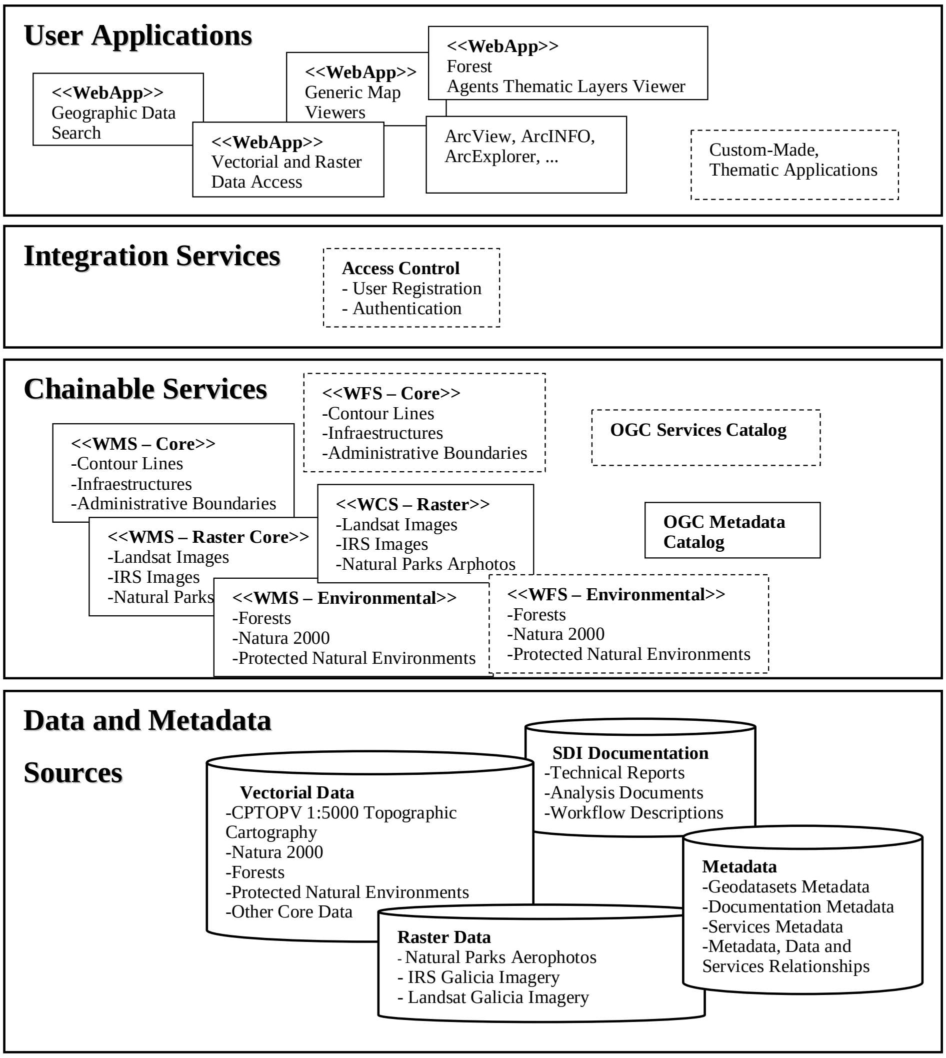

Figure 2. Galicia CMA SDI Architecture (taken from Béjar *et al.* (2003))

- 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. Applications are not Data Accessors in the SDI style, and neither are the 'User Applications' in the CMA SDI architecture because they do not access the 'Data Sources', they access the 'Services'.

Finally, it is important to notice that in this architecture, all components have the properties suggested for the SDI style (**name**, **type** and **types of data stored** for repositories) but the connectors do not have any.

4.2 *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, to facilitate the coordination of public sector organizations to collect, manage, distribute and reuse spatial data, designed according to INSPIRE principles (Cipriano and Garretti (2004)). This paper describes the components in the SITAD and gives the architecture diagram shown in figure 3. Although it is not indicated whether this diagram follows some existing architectural view type, it is sustained that it 'represents the presentation logic, the business logic and the data logic of the infrastructure' (p. 4). According to the architectural principles in Clements *et al.* (2003) 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, between the information portrayed in this diagram and the text of the paper there is enough to evaluate some elements of the architecture of the SITAD. These are the components described in the paper, matched, as far as it has been possible, to their equivalent type 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)) 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 says that data are accessed via on-line services and served to clients, what points that there must be connectors between data and services (at least **Data Reading**) and between services and clients (**Request/Reply**). This also probably 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, which seems to confirm this interpretation of what the text says.

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

4.3 *Architecture of the Northrine-Westphalia GDI*

The Geospatial Data Infrastructure Northrhine-Westphalia (GDI-NRW) is an initiative of the Land Northrhine-Westphalia, in Germany, that 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, as described in Brox *et al.* (2002). Besides 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 4. 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 ISO (2003). 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 4, 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.

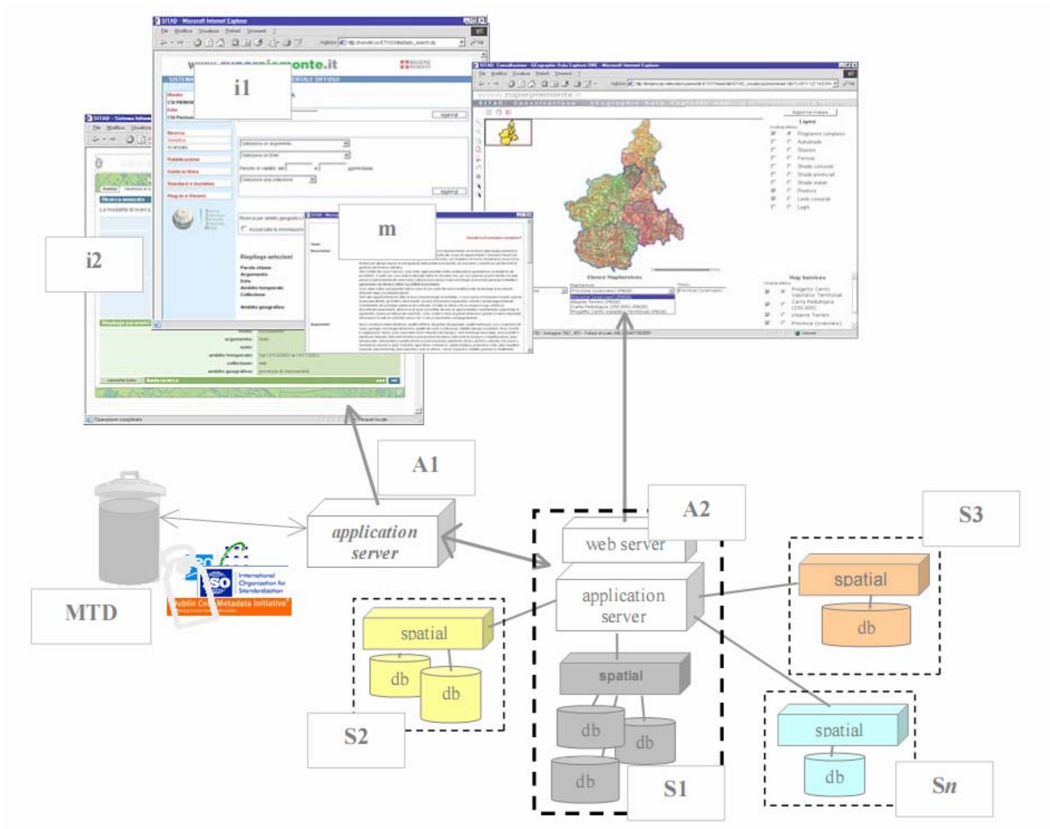

Figure 3. Architecture of SITAD infrastructure (taken from Cipriano and Garretti (2004) p. 5)

- 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 4, and following their function as explained in the paper, they all can be matched to some of the component types proposed in 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 be matched to those in the SDI style, though there are some aspects which 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 impor-

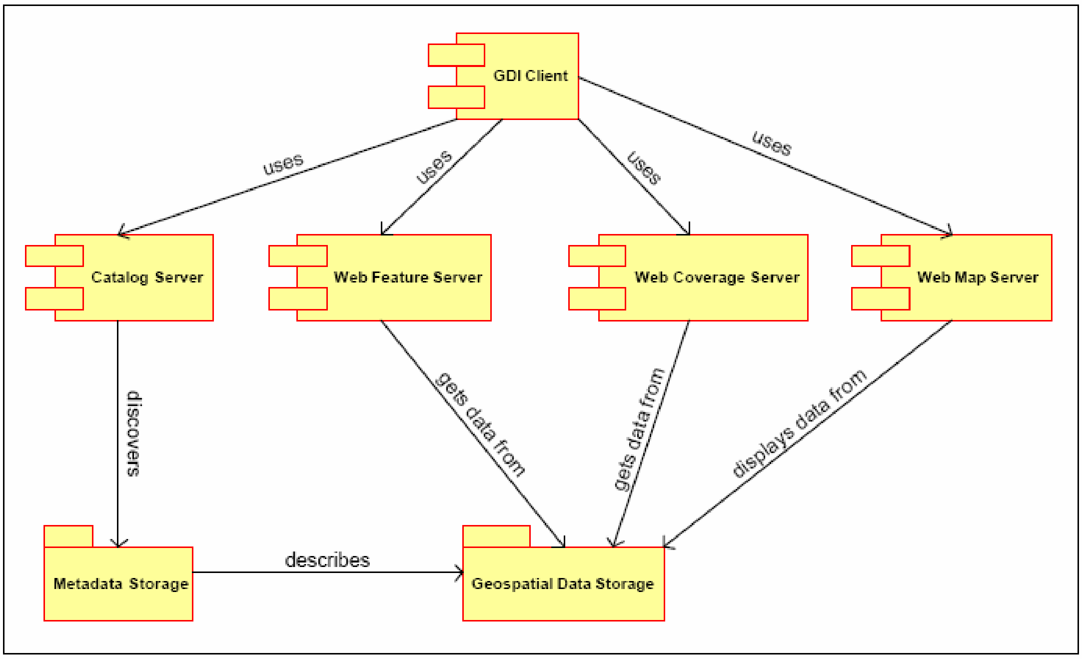

Figure 4. GDI Northrhine Westphalia Component Diagram (taken from Brox *et al.* (2002), p. 32)

tant 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 architectures; 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 much like the one discussed in the previous point.

From the text of the paper and figure 4, 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 on the meaning of this connector, but most probably it is not saying that the Catalog Server needs to discover where are 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, that 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 a different kind of connector from the others shown, because it does not show a connection between components: its place would probably be another diagram, with a different view type.

Regarding 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; it could thus 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, for the components and the connectors, is their **type**. As it is an architectural model which defines component types, more than an architectural view which would include components, this is the only property which makes clear sense.

5 Application of the the SDI style

This work 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 Clements *et al.* (2003) pp. 317-322 for some guidelines) far from the intention of this paper; 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; UML stereotypes will be used where needed to clarify their types (i.e. for the connectors). 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. As this architecture has been found to extend some of the component types in the SDI style, figure 5 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: how could have been documented a view of the architecture of the Galicia CMA SDI if the SDI style had been followed?

5.1 *Galicia CMA architecture view following the SDI Style*

Figure 6 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 a name and the stereotypes indicate their type. The question is, which are the differences between this diagram, and the one shown in figure 2?

- The meaning of this diagram is better defined, once the SDI style is known. Most component and connector types, all but 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

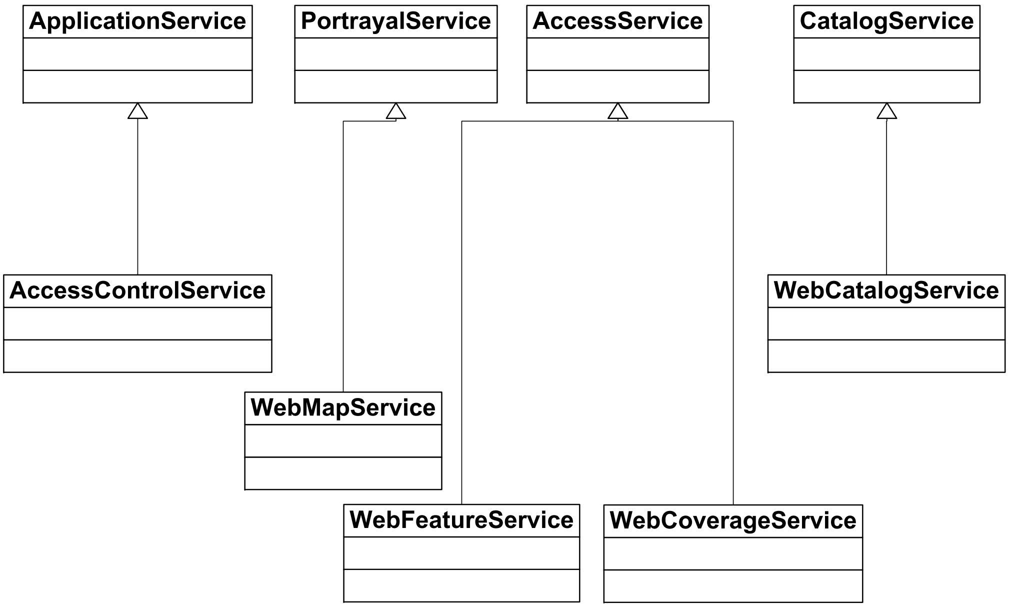

Figure 5. Galicia CMA SDI component types which extend those in the SDI Style

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: 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).

A final consideration: as the SDI style does not specify a graphical notation, a UML object diagram has been used. This kind of diagram is easy to explain, because UML is well-known, but a little more work would probably be required to make it more readable.

6 Conclusions

This work 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 inside 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, 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 that 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 in 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 is an SDI and which is its architecture. 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

(Draft) International Journal of Geographical Information Science, 2009, vol. 23, nº 3, p. 271-294.

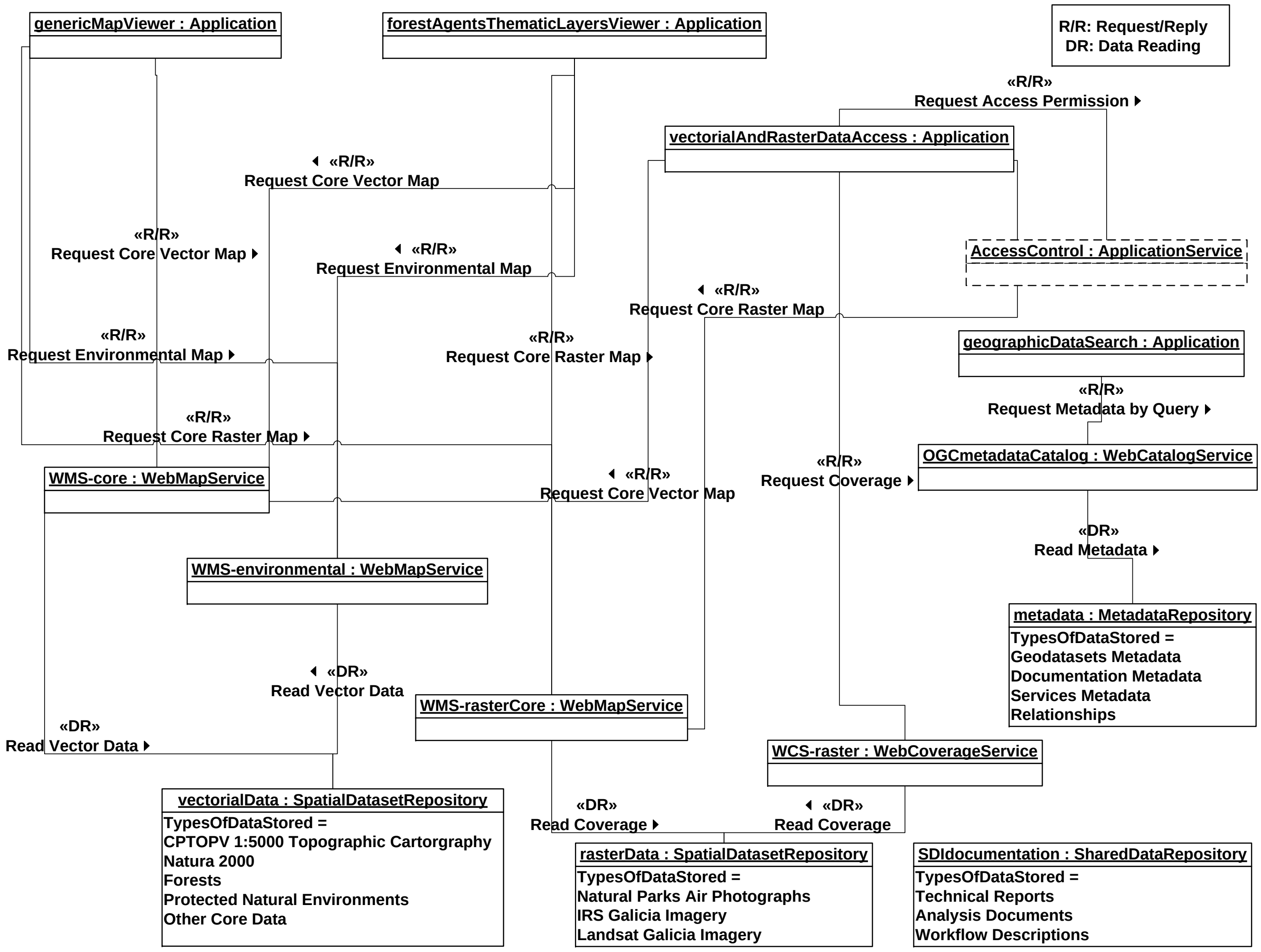

Figure 6. Galicia Architecture View following the SDI Style

implicitly, but a system architect may extend it to address specific necessities of her 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.

Another consideration must be done: there is a refinement of the SDI style that could have been considered: using 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, because OGC and ISO specifications are referenced in most, if not all, SDI related documents, a more abstract approach was decided 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 standardized services (i.e. an OGC SDI style), but which does not force them.

To finish this 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, that can be used to analyze certain properties, i.e. performance, of a software system (as explained by Clements *et al.* (2003)). The second issue is that besides distributed geographic information systems, as have been mainly considered here, SDIs are also Information Infrastructures. From this point of view their architecture can not be designed, but 'cultivated' (Georgiadou (2006)), so a necessary advance would be analyzing the architectural properties of an SDI that does not have, and can not have, a 'chief architect'.

References

AST Working Group, 2002, INSPIRE Architecture and Standards Position Paper. Technical report, INSPIRE Working Groups.

Bernard, L. and Craglia, M., 2005, SDI - From Spatial Data Infrastructure to Service Driven Infrastructure. In *Proceedings of the First Research Workshop on Cross-learning on Spatial Data Infrastructures and Information Infrastructures*, April, Enschede, the Netherlands Retrieved December 15, 2006 from http://gi-gis.jrc.it/ws/crosslearning/papers/PP Lars Bernard - Max Craglia.pdf.

Bernard, L., Einspanier, U., Lutz, M. and Portele, C., 2003, Interoperability in GI Service Chains - The Way Forward. In *Proceedings of the 6th AGILE Conference on Geographic Information Science*, April, Lyon, France, pp. 179–187.

Bernard, L., Kanellopoulos, I., Annoni, A. and Smits, P., 2005, The European geoportal - one step towards the establishment of a European Spatial Data Infrastructure. *Computers, Environment and Urban Systems*, **29**, 15–31.

Béjar, R., Gallardo, P., Gould, M., Muro-Medrano, P.R., Nogueras-Iso, J. and Zarazaga-Soria, F.J., 2004, A High Level Architecture for National SDI: the Spanish Case. In *Proceedings of the 10th EC-GI & GIS Workshop. ESDI: The State of the Art*, June 23-25 (Warsaw, Poland: Institute for Environment and Sustainability (IES), Joint Research Centre (JRC)), p. 6.

Béjar, R., Latre, M.Á., Gould, M. and Muro-Medrano, P.R., 2003, GIS COTS Integration in an SDI Software Architecture, a Study Case in the Galicia Region SDI. In *Proceedings of the Beiträge zu den Münsteraner GI-Tagen*, June, Munster, Germany, pp. 91–103.

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

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

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

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

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

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

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 Retrieved January 31, 2007 from http://register.consilium.europa.eu/pdf/en/06/st03/st03685.en06.pdf 2007.

Evans, J.D. (Ed.) , 2003, *A Geospatial Interoperability Reference Model*, 1.1 (FGDC Geospatial Applications and Interoperability Working Group) (Retrieved January 31, 2007 from http://gai.fgdc.gov/girm/v1.1/).

GeoConnections, 2005, The Canadian Geospatial Data Infrastructure Architecture Description Version 2.0. Technical report Retrieved January, 31 from http://www.geoconnections.org/publications/tvip/arch_E/CGDI_Architecture_final_E.pdf.

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

Georgiadou, Y., Puri, S.K. and Sahay, S., 2005, 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**, 1113–1130.

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

Harrison, J., 2002, OGC Web Services. Geoprocessing and the New Web Computing Paradigm. *Geoinformatics*, pp. 18–21.

Hofmeister, C., Nord, R. and Soni, D., 2000, *Applied Software Architecture* (Addison-Wesley).

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

ISF Working Group, 2002, Implementing Structures and Funding Position Paper. Technical report, INSPIRE Working Groups.

ISO, 2003, *Geographic Information - Services*, International standard; ISO 19119 (Genève, Switzerland: International Organization for Standardization).

Kruchten, P., Obbink, H. and Stafford, J., 2006, The Past, Present and Future of Software Architecture. *IEEE Software*, **23**, 22–30.

Kuhn, W., 2005, Geospatial Semantics: Why, of What and How?. In *Journal on Data Semantics III*, S. Spaccapietra (Ed.), 3534 of *Lecture Notes in Computer Science, LNCS* (Springer), pp. 1–24.

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

Lemmens, R., Wytzisk, A., de By, R., Granell, C., Gould, M. and van Oosterom, P., 2006, Integrating Semantic and Syntactic Descriptions to Chain Geographic Services. *IEEE Internet Computing*, **10**, 42–52.

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

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

Putman, J.R., 2000, *Architecting with RM-ODP* (Upper Saddle River, New Jersey: Prentice Hall).

Whiteside, A. (Ed.) , 2005, *OpenGIS web services architecture description*, no. OGC 05-042r1 In: OpenGIS Discussion Paper (Open Geospatial Consortium Inc.).

Wytzisk, A. and Sliwinski, A., 2004, Quo Vadis SDI?. In *Proceedings of the 7th AGILE Conference on Geographic Information Science*, May, Heraklion, Greece, pp. 43–49.