Universitat Jaume I
Departament de Llenguatges i Sistemes Informàtics

IMPROVING RESOURCE AVAILABILITY FOR GEOSPATIAL INFORMATION INFRASTRUCTURES

Ph. D Dissertation

Laura Díaz Sánchez

Supervisors:
Dr. **Michael Gould Carlson**
Dr. **Carlos Granell Canut**
Dr. **Joaquín Huerta Guijarro**

Castellón, May 2010

IMPROVING RESOURCE AVAILABILITY FOR GEOSPATIAL INFORMATION INFRASTRUCTURES

Laura Díaz Sánchez

Dissertation submitted to the Department of Languages and Computer Systems in partial fulfillment of the requirements for the degree of Doctor by the University Jaume I

Supervisors:
Dr. **Michael Gould Carlson**
Dr. **Carlos Granell Canut**
Dr. **Joaquín Huerta Guijarro**

Castellón, May 2010

This research has been partially supported by European project AWARE, the European project EuroGEOSS and by the CENIT España Virtual project through the Instituto Geográfico Nacional (IGN).

To my family.

Departamento de Lenguajes y Sistemas Informáticos

MEJORAS EN LA DISPONIBILIDAD DE RECURSOS EN INFRAESTRUCTURAS DE INFORMACIÓN GEOESPACIAL

Tesis doctoral
Laura Díaz Sánchez
Director: Dr. Michael Gould Carlson, Dr. Carlos Granell Canut,
Dr. Joaquín Huerta Guijarro

Castellón, Mayo 2010

Los sistemas de información construidos mediante servicios distribuidos basados en estándares se han convertido en el paradigma de computación adoptado por defecto por la comunidad geoespacial para la construcción de infraestructuras de información. Diferentes disposiciones gubernamentales, como la Directiva Europea INSPIRE, recomiendan normas para compartir recursos (por ejemplo, datos y procesos) con el objetivo de mejorar el estudio del medio ambiente (y otros campos similares) y la toma de decisiones.

La mayoría de las infraestructuras de información geoespacial (IIG) atienden necesidades básicas tales como acceso a datos, visualización y descarga, sin embargo, tienen escasos enlaces a servicios de geoprocesamiento. Además, la mayoría de infraestructuras geoespaciales se han construido siguiendo un enfoque del tipo *top-down*, donde sólo a los proveedores oficiales (por lo general las organismos públicos) se les permite desplegar y mantener recursos. Debido a que los mecanismos para desplegar e integrar recursos en estas infraestructuras son tecnológicamente complejos, existe una muy limitada participación por parte de los usuarios, provocando a la larga la escasez de recursos actualizados.

Para hacer frente a estas limitaciones, se presenta en este trabajo, una arquitectura distribuida basada en los principios de INSPIRE y ampliada con un componente llamado ServiceFramework. Este componente proporciona capacidad de procesamiento, ofreciendo una funcionalidad existente en forma de servicio estándar de procesamiento. También mejora la integración ad hoc y el despliegue de recursos de información geoespacial dentro de la infraestructura. El ServiceFramework trata la necesidad de mejorar la disponibilidad de recursos de datos geoespaciales, proporcionando mecanismos para generar de forma automática servicios compatible con INSPIRE.

Estas contribuciones son evaluadas en dos escenarios diferentes dentro de dos proyectos europeos. En el proyecto AWARE, se demuestra cómo un grupo de hidrólogos puede beneficiarse del acceso a las infraestructuras geoespaciales donde pueden compartir capacidades de procesamiento. En el proyecto EuroGEOSS, demostramos cómo mejorar la disponibilidad de los recursos geoespaciales para analizar un escenario forestal. Se prevé que estas nuevas metodologías, pueden aumentar la participación de usuarios más expertos en la creación de infraestructuras de información geoespacial, lo que aumenta la utilidad y el valor de estas infraestructuras.

Objeto y objetivos de la investigación

Los expertos en medio ambiente y otros usuarios geoespaciales necesitan acceder no sólo a recursos de datos sino también a servicios de geoprocesamiento para la generación de nueva información. Para ello, es necesaria una mejor conexión de las IIG a funcionalidades más sofisticadas, como el geoprocesamiento, para lograr una mayor interacción entre los datos disponibles. En el escenario actual, el resultado es que las infraestructuras geoespaciales son poco utilizadas, debido a los siguientes puntos:

- La falta de funcionalidad en línea, que debe ser proporcionada por otros medios, como aplicaciones de escritorio.
- Las IIG son dinámicas y participativas (Mansourian et al., 2005) y requieren un mantenimiento continuo. Los mecanismos de despliegue e integración de nuevos recursos son complejos y limitan las posibles contribuciones de los usuarios.

Por lo tanto, como objetivo proponemos que para mejorar las infraestructuras geoespaciales, debemos abordar los siguientes retos:

- Mejora de la IIG para conectar a herramientas de procesamiento en forma de servicios estándar para ejecutar análisis y modelos. Esto permitiría a los usuarios procesar y preparar los datos para extraer la información necesaria en cualquier situación.
- Desarrollo de herramientas para gestionar la persistencia de los recursos de datos. Por ejemplo, datos locales y/o resultados de ejecuciones de modelos se utilizan exclusivamente durante la ejecución de los modelos y no están disponibles en la IIG. Esta limitación (es decir, la falta de disponibilidad de recursos) impide que otros usuarios aprovechen estos datos para la validación y calibración en sus propios experimentos.

El desarrollo de herramientas para ayudar a los usuarios a integrar recursos en la IIG les permitiría convertirse en actores más activos en la construcción y mantenimiento de la infraestructura, dando lugar a una IIG más actualizada y útil.

Las IIG se basan en servicios web geoespaciales basados en estándares (Kiehle, 2006; Friis-Christensen et al., 2007), cuyos mecanismos de generación requieren capacidades tecnológicas avanzadas (personal cualificado, tecnologías geoespaciales y herramientas, etc.)

Por otra parte, las metodologías de construcción de las IIG siguen un enfoque *top-down*, donde los proveedores de datos oficiales despliegan sus recursos de acuerdo con políticas y servicios estándar. En este escenario se sigue un paradigma de proveedor-consumidor y limita el papel del usuario a la de un simple consumidor, lo que hace que contribuir a la IIG sea una tarea difícil.

La falta de metodologías de *bottom-up* impide la fácil integración e intercambio de información con otros interesados dificultando la rápida disponibilidad y visibilidad de los recursos geoespaciales. Esto afecta de forma negativa en ciertos campos, como la gestión de emergencias, donde el tiempo es un tema prioritario, incluso por encima de la calidad de los datos (Nayak y Zlatanova, 2008).

En este trabajo investigamos el uso de metodologías híbridas. Para abordar las cuestiones mencionadas proponemos una mejor integración de los enfoques de *bottom-up* y *top-down* donde la información generada por los usuarios, herramientas científicas y la información oficial (Dienel, 1989;. Jankowski,

2009) se pueden desplegar e integrar como componentes interoperables en la IIG.

La idea clave de nuestra propuesta consiste en cambiar el papel de los usuarios, transformándolos de puros consumidores en participantes activos que pueden proporcionar e integrar nuevos recursos.

Planteamiento y metodología

Estado Del arte

Previamente a la exposición de la nuestra propuesta, hacemos una revisión del estado del arte relevante para nuestro trabajo. Esta sección trata diferentes problemas de interoperabilidad, analiza los enfoques de las IIG y servicios web, además de trabajos relacionados con la integración de los recursos geoespaciales y reutilización (tanto recursos de procesamiento como de datos). Además, describe metodologías utilizadas para la construcción de IIG. Para finalizar, se analizan algunas aplicaciones y herramientas relacionadas con nuestros escenarios y casos de uso.

Los ingredientes necesarios para el éxito del trabajo colaborativa y multidisciplinar, en el escenario de las IIG son: (1) nuevos estilos arquitectónicos opuestos a soluciones centralizadas y aisladas, (2) la integración del procesamiento distribuido, y (3) integración de los conocimientos del usuario.

Arquitectura

Este capítulo está dedicado a analizar los requisitos de la IIG, especialmente en el ámbito medioambiental, y proponer una arquitectura conceptual con los componentes necesarios para cumplir estos requisitos.

Además de los requisitos generales como el descubrimiento, visualización y descarga, hemos identificado las siguientes los requisitos a abordar:

- Acceso a procesamiento distribuido en IIG para modelización, extracción y preparación de la información.
- Mantenimiento y construcción asistida de IIG. Para poder ejecutar aplicaciones distribuidas sobre IIG es importante una buena disponibilidad de recursos interoperables.

- Los usuarios no participan activamente en las IIG actuales debido a los complejos mecanismos de despliegue y a las metodologías de construcción tradicionales que siguen una metodología top-down, donde sólo los proveedores oficiales pueden integrar nuevos recursos.

La arquitectura básica propuesta sigue los principios de INSPIRE y se ha ampliado con nuestra contribución: el ServiceFramework. En esencia, este componente actúa como un generador de servicios.

Los componentes de esta arquitectura se describen en el capítulo 4, donde nos centramos en los componentes y mecanismos para añadir recursos (herramientas y datos) integrados como servicios estándares para mejorar su accesibilidad y visibilidad. El capítulo 5 describe los casos de uso en los que vamos a evaluar esta arquitectura y sus componentes para comprobar los resultados y limitaciones.

Desarrollo

La tendencia actual en los sistemas de información es la migración de sistemas monolíticos a entornos distribuidos. Los IIGs están experimentando el mismo cambio (Bernard, 2003).

La interoperabilidad se supone que debe ser garantizada por una serie de esfuerzos en la generación de especificaciones, como las normas ISO / TC 211 y OGC en la comunidad geoespacial (Bernard, 2003) y otros marcos de interoperabilidad como INSPIRE. La adopción de los servicios estándar y su despliegue en la capa de middleware de la IIG hace que sea factible la implementación de aplicaciones distribuidas geoespaciales.

En nuestro contexto, hacemos una distinción entre dos tipos de servicios: servicios de datos y servicios de procesamiento. Esto significa que es necesario envolver: (1) datos e información que sean expuestos por los servicios de datos estándar y (2) herramientas (procesos, algoritmos) que se exponen como servicios estándar de procesamiento estándar.

De este modo, describimos, por un lado cómo envolver herramientas y funcionalidades para generar servicios de procesamiento que implementan el interfaz estándar de OGC Web Processing Service (WPS). Por otro lado, describimos como mejorar la disponibilidad de recursos de datos geoespaciales mediante la generación automática de servicios para visualizar y descargar datos conforme a la directiva INSPIRE. Por último tratamos de mejorar la visibilidad de los recursos mediante el registro automático de estos servicios en Catálogos de Servicio.

Metodología

Se ha realizado un estudio del arte y se han señalado ciertas cuestiones por resolver y que pretendemos abordar en este trabajo. Para el desarrollo de nuestra propuesta seguimos la metodología de desarrollo iterativo e incremental, donde se describe el análisis, el diseño y implementación. En el último paso de nuestra metodología (es decir, la evaluación) se describe un caso de uso para mostrar nuestra solución en el contexto de dos modelos hidrológicos conocidos en el marco del proyecto AWARE. Nuestro enfoque es lo suficientemente genérico como para que pueda aplicarse a otras disciplinas y ámbitos, como vemos en un segundo caso de uso proporcionado por el proyecto EuroGEOSS.

Aportaciones originales

Las principales aportaciones de este trabajo son la investigación y propuesta de una metodología y una solución tecnológica para ayudar a los usuarios a trabajar de forma más eficiente participando más activamente en la GII. El ServiceFramework es la implementación de la solución tecnológica propuesta para los problemas mencionados.

En primer lugar se propone una arquitectura y una metodología para diseñar e implementar servicios de procesamiento para las IIG. Con ejemplos, nuestro trabajo muestra cómo la exposición de instrumentos científicos a través de servicios web estándares tiene la ventaja de eliminar la necesidad y mantenimiento de software y capacidad de cómputo en los ordenadores locales, y proporciona acceso remoto a datos y herramientas necesarias para la ejecución de modelos científicos. El componente principal de esta contribución es el ProcessWrapper.

En segundo lugar, el componente DataWrapper dentro del ServiceFramework asiste a los usuarios en la construcción y el mantenimiento de estas infraestructuras mediante el despliegue de nuevos recursos de información como servicios estándares compatibles con la directiva INSPIRE. Nuestro objetivo es demostrar cómo podemos alcanzar un factor multiplicativo alto en el intercambio de recursos al ocultar la complejidad de los mecanismos de implementación y despliegue en IIG. Al proporcionar mecanismos más simples de participación, el usuario puede convertirse en un proveedor y participar en el desarrollo de IIG.

Por último, el ServicePublisher registra de forma automática los servicios generados en los Servicios de Catálogos disponibles en la IIG mejorando la visibilidad de los recursos.

Para lograr nuestro objetivo, hemos desarrollado herramientas que tomando datos geoespaciales y herramientas facilitados por los usuarios, generan y despliegan automáticamente servicios geoespaciales basados en interfaces estándar. Esta metodología tiene el valor añadido de que los datos científicos y las herramientas servidos a través de servicios Web estándar pueden ser reutilizados en otros escenarios.

Conclusiones

Usuarios, científicos y técnicos generan herramientas y grandes volúmenes de información. Las IIG les proporcionan componentes estándar para encontrar y acceder a recursos distribuidos de una manera interoperable para facilitar su trabajo. Sin embargo, estas infraestructuras no tienen en cuenta los mecanismos necesarios para facilitar la colaboración de usuarios en el mantenimiento de estas infraestructuras. Esto implica que aún existen muchos recursos científicos que no están disponibles para ser compartidos en estas infraestructuras.

Hemos descrito una arquitectura y hemos propuesto una extensión con un componente llamado el ServiceFramework para ayudar a los usuarios de IIG en el despliegue e integración de sus recursos, mediante la generación automática de servicios estándar para servir estos recursos. Hemos tratado dos puntos principales para compartir recursos: en primer lugar el ocultamiento de la tecnología a los usuarios para poder añadir, de forma masiva, nuevos recursos a plataformas distribuidas y en segundo lugar, envolver estos recursos como servicios estándar, en nuestro caso siguiendo especificaciones de OGC recomendadas por INSPIRE, garantizando la interoperabilidad de los nuevos recursos.

Hemos hablado de aumentar la disponibilidad, visibilidad de recursos y la interoperabilidad, pero debemos remarcar que hablamos de interoperabilidad sintáctica alcanzada mediante el uso de estándares OGC recomendados por la Directiva INSPIRE.

Futuras líneas de investigación

- Seguridad

Los mecanismos para permitir a los usuarios publicar sus recursos conducen a problemas de seguridad o privacidad. Las cuestiones de seguridad aún no son tomadas en cuenta y es algo a considerar en el futuro como la validación de usuarios y sus derechos.

- OWS-T: interfaces transaccionales

ServiceFramework utiliza una implementación concreta de OGC WMS, WFS, WCS y su protocolo para exponer recursos como servicios estándar. Para no tener dependencias y poder migrar a otras implementaciones, en un futuro sería útil investigar la posibilidad de las operaciones transaccionales estándar.

- Interoperabilidad semántica

Para alcanzar niveles aceptables de integración ad hoc de datos espaciales y procesos es necesario futuras investigaciones sobre la interoperabilidad semántica a esta escala.

- Simbología

ServiceFramework permite al usuario simplemente elegir un color para la visualización de datos. En un futuro se investigara la generación de un conjunto de estilos automáticos para que los usuarios puedan compartirlo mediante su publicación en OGC WMS, por ejemplo, generando OGC SLD.

- Metadatos

Actualmente sólo generamos un conjunto mínimo de elementos de metadatos para la descripción de servicios y publicación en un catálogo de servicios. Como ya hemos mencionado, existen paquetes de software capaces de generar automáticamente algunos elementos de metadatos. Estas aplicaciones podrían ser integradas dentro del ServiceFramework en el futuro. Otra línea de investigación es ver como generar metadatos consultando a los servicios donde hemos desplegado los recursos mediante el uso de las interfaces OGC que proporcionan información sobre los recursos que sirven.

Abstract

Information systems built using standards-based distributed services have become the default computing paradigm adopted by the geospatial community for building information infrastructures. Government mandates such as the INSPIRE European Directive recommend standards for sharing resources (e.g. data and processes) with the goal of improving environmental (and related) decision-making.

Most geospatial information infrastructures address basic needs such as data access, visualization and download, however they have limited links between data and geoprocessing services. Also, most geospatial infrastructures have been built following a top-down approach where official providers (most commonly mapping agencies) are permitted to deploy and maintain resources. Because the mechanisms to deploy resources in these infrastructures are technologically complex, there has been very limited participation from users, resulting in a scarcity of deployed resources.

To address these limitations, we present a distributed architecture based on INSPIRE principles and extended with a ServiceFramework component. This component assists users by adding processing capabilities, wrapping existing functionality as a standard Web Processing Service. It also improves ad hoc integration and deployment of geospatial data resources within an infrastructure. The ServiceFramework further addresses the need to improve the availability of geospatial data resources by providing mechanisms to assist users in wrapping resources to generate INSPIRE-based services.

These contributions are evaluated within two different EU project scenarios. In the AWARE project, we demonstrate how hydrological scientists benefit from access to geospatial infrastructures where they can share processing capacities. In the EuroGEOSS project, we demonstrate how improved availability of geospatial resources help collaborative research teams to analyze forestry scenarios. It is anticipated that new methodologies such as the one described here will increase the participation of more expert users in the creation of geospatial information infrastructures, thereby increasing the usefulness and value of these infrastructures.

Acknowledgments

First of all I would like to thank my advisor Michael Gould for introducing me to the geospatial world; for his contagious enthusiasm and for his support even when distance came into play. Special thanks to my co-advisor, Carlos Granell, for helping me complete the writing of this dissertation as well as for the challenging research chats by the blackboard. Special thanks as well to my co-advisor, Joaquín Huerta, for helping me find always a way to move forward.

Besides my advisors, I would like to thank the rest of my expert thesis committee: Sisi Zlatanova for her valuable time and great comments, Stefano Nativi for his encouragement and good questions and Pedro Muro for reporting constructive comments and reviewing my work on a very short notice.

I am very grateful to Christine Lam and Evan Lue, from the Department of Geography at the University of Southern California, for their revisions and helpful comments which help me complete and improve the writing of my dissertation.

To my colleagues and friends for their interesting comments always presented as lovely conversations in many different scenarios. To Salvador Bayarri for his inspiring talks, to Egon Amade for his priceless time and effort to help me improve my work and Jose Gil for giving to the ServiceFramework a beautiful face. And of course to my colleagues from the LSI department, especially to those from CEVI for always being available to help and contribute to a very nice working environment.

I would like to thank all my friends for being there and enriching my life in so many different ways.

Finally, I want to thank and dedicate this work to my family, specially my parents who always support me unconditionally and show me which kind of person I would like to become. To my sisters Eva and Raquel and my brother Iván for always listening to my complaints and believing in me. To Ana for her patience and attention, for helping me see the nice side of everything, and for her creativity which makes everything look great.

Index

Abstract XI

1. Introduction **1**

1.1 Motivation 2
1.2 Problem 4
1.3 Contributions 6
1.4 Context 7
1.4.1 AWARE Project 7
1.4.2 EuroGEOSS Project 8
1.5 Structure of thesis 8

2. Related work **9**

2.1 Interoperability issues 9
2.1.1 Scope 10
2.1.2 Interoperability levels in information systems 10
2.1.3 Interoperability in geospatial information infrastructures 11
2.1.4 Syntactic interoperability of geospatial services 12
2.1.5 Interoperability limitations in geospatial information infrastructures 13
2.1.6 Cross-platform interoperability 13
2.2 Geospatial information infrastructures (GII) 14
2.2.1 Service Oriented Architectures (SOA) and Spatial Data Infrastructures (SDI) 14
2.2.2 International initiatives 17
2.2.3 European Initiatives 18
2.3 Geospatial services 20
2.3.1 OGC Web Data Service specifications 22
2.3.2 OGC Web Data Service implementations 23
2.3.3 OGC Web Processing Service specification 24
2.3.4 OGC Web Processing Service Implementation 26
2.4 Geoprocessing in Spatial Data Infrastructures 27
2.5 Wrapping and sharing geospatial resources as web services 28
2.6 User generated information 30
2.7 Environmental applications 32
2.8 Summary 35

3. Architecture 37
3.1 Introduction 37
3.2 Analysis of requirements 37
3.2.1 Spatial user requirements 38
3.2.2 SDI Building requirements 39
3.2.3 Domain applications requirements 41
3.3 Architecture Principles 43
3.4 Proposed Basic Architecture 44
3.5 Proposed extended Architecture 46
3.6 Summary 48

4. ServiceFramework: Geospatial Resources Availability 49
4.1 Introduction 49
4.2 Methodology 51
4.3 Processing resource availability 52
4.3.1 Analysis 52
4.3.2 Design 54
4.3.3 Implementation 57
4.4 Information resources availability 70
4.4.1 Analysis 71
4.4.2 Design 73
4.4.3 Implementation 74
4.5 Service Publisher 85
4.5.1 Analysis 86
4.5.2 Design 87
4.5.3 Implementation 88
4.6 Summary 93

5. Evaluation: Experiments on the Use cases 95
5.1 Experiments on the AWARE project: Generation and deployment of processing services to run scientific models. 95
5.1.1 Analysis 95
5.1.2 Architecture 97
5.1.3 Design 99
5.1.4 Implementation 105
5.1.5 AWARE Hydrological models execution 105
5.2 Experiments on the EuroGEOSS project: Improving data availability to perform a protected areas damage assessment. 110
5.2.1 Analysis 111
5.2.2 ServiceFramework integration in EuroGEOSS architecture115
5.2.3 Implementation 119
5.2.4 EuroGEOSS Protected areas damage assessment 121

6. Conclusions.......... 131
6.1 Processing resources availability.......... 131
6.2 Data resource availability.......... 133
6.3 Resource visibility.......... 137

Bibliography.......... 139

Annex A: Aware Processing Services Design.......... 157

Annex B: Aware Processing Services Implementation.......... 181

Index of figures

Figure 1. Contributions graphical overview: ServiceFramework conceptual components. 7
Figure 2. Interoperability and standards stack (extracted from "Standards into Action" (Nebert, 2005) 12
Figure 3. GEOSS scope (GEOSS, 2007) 17
Figure 4. INSPIRE technical architecture (INSPIRE, 2007) 20
Figure 5. OGC Service tiers (OGC Reference Model) 23
Figure 6. Data Integration and Processing OGC Services 24
Figure 7. WPS interface UML diagram 25
Figure 8. Synchronous interaction between a WPS-compliant client and a WPS service instance 25
Figure 9. GEOSS technological use cases 38
Figure 10. Providers-Consumers paradigm sequence diagram 39
Figure 11. Users as providers-consumers in a bi-directional environment 40
Figure 12. Extended GEOSS technological use cases 41
Figure 13. Steps to generate and share information in emergency management: (i) and (ii) resource access, generation and deployment; (iii) and (iv) sharing and collaborative decision-making and actuation 41
Figure 14. Geospatial resource life cycle in a distributed environment 42
Figure 15. Conceptual architecture for distributed applications 44
Figure 16. Extended architecture for sharing user generated resources 47
Figure 17. ServiceFramework integrating processing services generation, data services generation and service publication 50
Figure 18. Three software development methodologies (Extracted from Wikipedia) 51
Figure 19. Iterative and incremental process (extracted from Wikipedia) 52
Figure 20. Activity diagram: Design and implementation steps for integrating existing or new software as WPS in GII 55
Figure 21. Adding a new process using 52North OGC WPS implementation ... 59
Figure 22. Example OGC WPS *GetCapabilities* 60
Figure 23. Example OGC WPS describeProcess 61

Figure 24. Example OGC WPS Execute request 62
Figure 25. Example of an OGC WPS execute response....................................62
Figure 26. Extended architecture with details on the components involved in increasing the availability of processing resources 63
Figure 27. WPSAPI class diagram 65
Figure 28. Process Wrapper Components diagram 67
Figure 29. Process Wrapper delegator architecture..68
Figure 30. Example of decisions from abstract definition to implementation using Sextante WPS (extracted from AWARE use case) 69
Figure 31. Adapter pattern. The left side depicts generic patterns and the right side depicts the pattern implementation applied to semi-automatic migration from SEXTANTE algorithms to WPS. 70
Figure 32. Information generation process 72
Figure 33. Activity diagram: Activities in the design and implementation steps for integrating data resources as standard data services in GII 73
Figure 34. Architecture and components with details for data wrapping.........76
Figure 35. Conceptual components diagram 77
Figure 36. Data wrapper detailed components 78
Figure 37. Sequence diagram for data wrapping..80
Figure 38. ServiceFramework interface regarding the data wrapping 80
Figure 39. Data published in the Geoserver instance in *ServiceFramework* 81
Figure 40. Retrieving data from uDIG OGC WMS and WFS client 81
Figure 41. uDIG screenshot shows how published data in OGC services can be retrieved by interoperable OGC clients 82
Figure 42. Sequence diagram for user-generated information wrapping.........84
Figure 43. Overview on the general steps in the design and implementation for integrating data resources as standard data services in SDI 87
Figure 44. Service Publisher Component Diagram 91
Figure 45. Sequence diagram of publishing OGC Services in Service Catalogues with the ServiceFramework 92
Figure 46. Service Publisher interface (proof of concept) 93
Figure 47. General use case 96
Figure 48. Area calculation use case 97
Figure 49. AWARE architecture based on the proposed architecture described in Chapter 3...98
Figure 50. Elevation Zones process workflow..101

Figure 51. Service perspectives: types, design and implementation. Service types is about INSPIRE vs. AWARE service types. Design is about granularity (instances) and interfaces. Implementation is about specific tools and programming languages. 102
Figure 52. User interface for retrieving metadata records from a catalogue service. 106
Figure 53. User interface to visually inspect temperature sensor data 107
Figure 54. UML activity diagram to calculate the elevation zones 108
Figure 55. User interface displaying the elevation curves for the watershed 109
Figure 56. Results with plots comparing calibrated, real and simulated discharges of the watershed. 110
Figure 57. Activity diagram with the tasks of the workflow of the selected use case 114
Figure 58. EuroGEOSS IOC 116
Figure 59. Details on EuroGEOSS Broker component 117
Figure 60. Proposed Architectural Overview for Forest Capability (blue) 118
Figure 61. EFFIS architecture with the relations with *ServiceFramework* 119
Figure 62. Service Framework Simplified UML diagram 120
Figure 63. User can upload local data to be integrated in the system 122
Figure 64. Visualization of datasets. 122
Figure 65. Visualization of datasets (II) 123
Figure 66. After uploading data, these are visualized on the viewer. 123
Figure 67. Protected areas dataset deployed as INSPIRE View and download Service type according to OGC standards 124
Figure 68. Burned areas dataset deployed as INSPIRE View and download Service type according to OGC standards (II) 125
Figure 69. Geoserver as OGC WMS-WFS-WCS implementation serving the uploaded datasets 125
Figure 70. ServiceFramework user interface lets the user run WPS by means of the ServiceConnector-WPS API 126
Figure 71. Visualization of process results 127
Figure 72. Deploying Processing results as INSPIRE Service types 128
Figure 73. ServicePublisher within the ServiceFramework lets the user publish WPS not yet available in OpenCatalogues 129
Figure 74. Data retrieval using gvSIG and UDig. 130
Figure 75. DataConversion conceptualization 157
Figure 76. Shp2GML abstract specification 158

Figure 77. Conceptualization of TopologyWPS....159
Figure 78. Area process abstract specification....160
Figure 79. Intersection operation. (a) polygon-polygon; (b) polygon-line....160
Figure 80. Intersection process abstract specification....161
Figure 81. Buffer operation. (a) polygon; (b) line....161
Figure 82. Buffer process abstract specification....162
Figure 83. MaxExtent operation. (a) polygon; (b) line....162
Figure 84. MaxExtent process abstract specification....163
Figure 85. Thiessen polygons....163
Figure 86.Thiessen process abstract specification....164
Figure 87. SnowPercentage process abstract specification....164
Figure 88. Conceptualization of SextanteWPS....166
Figure 89. CoordinateElevation process abstract specification....166
Figure 90. HypsometricElevation process abstract specification....167
Figure 91. StationsElevation process abstract specification....167
Figure 92. Classify algorithm according to three temperature ranges....168
Figure 93. Reclassify process abstract specification....168
Figure 94. Vectorize process abstract specification....169
Figure 95. ElevationZones process abstract specification....169
Figure 96. Elevation Zones functional chain....170
Figure 97. SRMIDLWPS conceptualization....171
Figure 98. SRMIDL process abstract specification....171
Figure 99. ChartWPS conceptualization....173
Figure 100. SRM Depletion Curves Plot....173
Figure 101. DepletionCurvesPlot process abstract specification....174
Figure 102. SRM Discharge Curves plot....174
Figure 103. DischargePlot process abstract specification....175
Figure 104. HBV runoff plot....175
Figure 105. HBVRunoffPlot process abstract specification....176
Figure 106. HBV Snow-Water-Equivalent plot....176
Figure 107. SensorData process abstract specification....177
Figure 108. Basin example in raster and vector format....178
Figure 109. Basin Abstract model....178
Figure 110. Example of a drainage basin divided in 4 elevation zones....179
Figure 111. Elevation Zone Abstract model....179

Index of tables

Table 1. OGC specifications to deploy data resources in SDI 75
Table 2. Services and processes used in the AWARE Application. 104
Table 3. DataConversionWPS service overview 157
Table 4. Topology WPS service overview 159
Table 5. Sextante WPS service overview... 165
Table 6. SRMIDLWPS service overview 170
Table 7. chartWPS service overview 172

1. Introduction

After thousands of years exploiting Earth's resources, we wonder whether we have done it in a sustainable way. Humankind needs to be aware of our vulnerability to natural disasters such as desertification, droughts, floods, and natural resource depletion as well as anthropogenic effects on the environment such as pollution.

Because awareness of environmental problems is increasing, environmental sciences are experiencing a forward-moving momentum as scientists aim to study and understand our biophysical environment. There are many challenges to understanding the Earth's behaviour from a multidisciplinary point of view (Goodchild 2008a; Craglia et al., 2008).

To face these challenges, scientists and decision-makers need global availability of current geospatial data and tools to extract accurate and useful information. In this sense Geospatial Information Systems (GIS) have become indispensable tools for accessing and organizing geospatial resources for environmental sciences, and providing a framework for multidisciplinary analysis (Ramamurthy, 2006).

Geospatial information is essential for achieving these challenges; the need for Earth Observation (EO) data to support the sustainable development of our planet was recognized by the 2002 World Summit on Sustainable Development (UN, 2002). And although the amount of geospatial data collected has increased significantly due to advances in data capture technologies (e.g. the National Aeronautics and Space Administration (NASA) identified nearly three thousand satellites currently orbiting the Earth) (Phillips, 1999; Ramamurthy, 2006), these resources are available from multiple sources and stored in multiple formats, and need to be organized in order for scientists and other users to access and exploit them [Gore, 1999].

Major natural disasters that occurred in the last few years such as the Indian Ocean earthquake in 2004 or Hurricane Katrina in 2005 raised several alarms. Particularly, the amount of spatial resources to manage disasters was insufficiently organized to provide an effective system of analysis and response.

1.1 Motivation

As with other scientific domains, geospatial and environmental sciences have made use of technological advances to improve the way of working. They rely on geospatial technologies to manage and exploit resources scattered across numerous agencies and in multiple formats (Phillips et al., 1999).

Multidisciplinary research teams require the support of suitable information infrastructures (II) that allow them to share distributed data and computing capabilities to achieve their objectives more efficiently (Hey and Trefethen, 2005).

In this geospatial domain, Geospatial Information Infrastructure (GII), also known as Spatial Data Infrastructures (SDI), is one of the best approximations to an eScience community. The capability of discovering, accessing and sharing a diversity of geospatial resources among a wide range of actors is being addressed by interconnected SDI nodes at different scales to build a global spatial II (Masser et al., 2007; Rajabifard et al., 2002).

The current trend in geospatial applications is to deploy SDI following the Service Oriented Architecture (SOA) paradigm based on access to resources via standard web services. SDI comprises a set of policies and standard activities promoting the creation of geospatial information services to assist diverse user communities in collecting, sharing, accessing and exploiting geospatial resources (Phillips et al. 1999; Nebert 2004; Masser, 2005; Bishop et al., 2000; Davis et al., 2009; Vandenbroucke et al., 2009).

Working with distributed applications on top of SDI increases interoperability, but the distribution of the information and the processes in different sources and applications have made the task of discovering and processing the data arduous (Scholten et al., 2008). Moreover, specialized users and particular domains require more advanced requirements like rapid access to up-to-date information, processing capabilities and collaborating with generated information and knowledge to be integrated in the II.

In the environmental domain, and particularly in the emergency management domain, good availability and rapid access to up-to-date information are very important since the first hours of response to a disaster are very critical for saving human lives and reducing damages (Diehl et al., 2006; NRC 2007, Zlatanova and Fabbri 2009; Zlatanova and Dilo, 2010). For instance, in the case of wildfire emergency, getting the information faster than the spread of the fire is a key issue. Having all the processes as well as the data well-described and published in a SDI will enhance the information delivery in the very first hours (Scholten et al., 2008).

Moreover, besides access to the right data at the right time, the data need to be presented it in an appropriate way (i.e. provide experts and decision makers with tailored output to ensure the understanding of the situation) (Almer et al., 2008; Brunner et al., 2009) as well as to let them run their scientific models to generate the required information. Therefore, there is a need for II to be processing-enabled and offer processing capabilities in an interoperable way.

During the execution of environmental applications as well as in emergency response situations, a lot of dynamic information is created after processing and preparing the data. This newly generated and up-to-date information is normally underused, since it is used for that very moment and normally not deployed in II in the form of a standard component. Consequently, further analysis of this information by other stakeholders is difficult and often not possible, in part because this information is not archived in a structured and interoperable way (Zlatanova and Dilo, 2010). Providing mechanisms to deploy this information as standard components would improve the availability of up-to-date resources, and thereby the maintenance of the geospatial infrastructures. This is a crucial aspect to understand the context of environmental applications used in emergency situations (Mansourian et al., 2005; Zlatanova et al., 2006; Rocha et al., 2005).

While SDI is the correct framework to organize geospatial resources in theory, these infrastructures are underutilized because many resources are not available or visible to a wide audience in practice, and this decreases user motivation. Therefore one of the main issues to address is the availability and visibility of these geospatial resources, where visibility and availability means to be able to find and access these resources deployed in these infrastructures in a structured and interoperable way.

Recent natural disasters such as the Indian and Chilean tsunamis, Hurricane Katrina, and forest fires in Greece and California demonstrated that difficulties still exist to efficiently access and work with geospatial resources in GII. The difficulties stem from the absence of sufficient available resources and a lack of collaboration and interrelation between different geospatial infrastructures and components.

In contrast with these cyber-infrastructures with standard components and complex deployment mechanisms, we are witnessing the consolidation of a new generation of the World Wide Web, in which the main features are user participation and greater usability. Tim O'Reilly (2005) popularized the evolving nature of the web by introducing the term 'Web 2.0.' The main innovation is a change in user role where end-users are not only information consumers but information providers, thereby establishing a bi-directional user-system relation where everybody can use the Web to create and share content.

The Geospatial Web, considered a natural extension of the World Wide Web, is also experimenting with this evolution towards GeoWeb 2.0 (Maguire, 2006) involving broader clients and web services (Kralidis, 2007). This technology aims the user to participate, for instance, in local decision-making (Jankowski, 2009)

An example of how useful user collaboration can be to generate and provide resources are the photos uploaded on sites (e.g. Flickr[i]) by the public during the California wildfires in 2007. These photos provided a quicker overview of the situation than information coming from traditional channels like mapping agencies. During Hurricane Katrina in 2005, Google Earth images were more useful than the United States Geological Survey (USGS) maps for rescue workers in describing the current situation (Nature, 2006; Budhathoki et al, 2008).

A more recent example is the earthquake in Haiti in January 2010. During the first weeks of the disaster we have witnessed that official geospatial resources were missing. Geospatial information to manage health and food distribution points were built by volunteers, using in most cases web tools to create and share this valuable information. This is possible thanks to technological advances and current devices like cell phones, Global Positioning System (GPS), etc., that allow humans to act as sensors, being able to build and publish content from the ground up (Goodchild, 2007a).

1.2 Problem

Despite the fact that the geospatial community has found in SDI the interoperable platform to share geospatial resources in the form of standard services, there are a few issues to be considered and that will be addressed in this dissertation.

Environmental experts and end users need to access computing and data resources, but also geoprocessing and generation of new information. In the SDI community, however, the connection to sophisticated geospatial capabilities like processing are still required to achieve a better interaction. The result is that geospatial infrastructures often are underutilized due to the following points:

- Lack of online functionality that must be supplied by other means such as desktop applications.
- SDIs are dynamic and multi-participant (Mansourian et al., 2005) and require continuous maintenance. Their complex deployment mechanisms limit the possible contributions of expert users who do not form a part of the top-down structure.

Therefore, we propose that to extend geospatial infrastructures in the form of SDI, we must attend to at least the following challenges:

- *Improvement of SDI to connect to sophisticated processing tools in the form of standard services to run and share analysis and models.* This would allow users to process and prepare data to extract the information required in any situation.
- *Development of tools to manage persistence of data resources like the outputs generated by models.* For instance, forestry models compute fire assessment maps, but they are used exclusively during the execution of the models and are not available in an accessible infrastructure. This limitation (i.e. lack of resource availability) prevents other users from harnessing such maps as resources for validation and calibration in their own experiments.

The development of tools to assist users in aggregating resources into the SDI would allow them to become more active stakeholders in SDI building and maintenance and, in theory, should result in a more up-to-date and more useful infrastructure.

The main issues are that operational SDI nodes rely on geospatial standard-based web services (Kiehle, 2006; Friis-Christensen et al., 2007), and the mechanisms to deployment these services in such infrastructures require to handle greater complexity (Béjar et al, 2009a) as the infrastructure grows. Furthermore these mechanisms require advanced technological capabilities (skilled personnel, geospatial technologies and tools, etc.).

Moreover, SDI building methodologies traditionally follow a top-down approach, where official data providers, like public administration, deploy their resources according to some policies and standard services. This scenario follows a provider-consumer paradigm and limits the end-user role to the one of a simple consumer which makes contribution to the knowledge repository a difficult task.

The lack of bottom-up capabilities impedes the rapid integration and sharing of crucial information with other stakeholders and impedes fast availability and visibility of geospatial resources. This is negatively affecting certain domains like emergency management where time is an important issue and speed must be prioritized over quality (Nayak and Zlatanova, 2008).

The use of hybrid methodologies is investigated in this work. To address the mentioned issues, we propose to better integrate bottom-up approaches and top-down methodologies where user generated information, scientific tools and official information (Dienel, 1989;. Jankowski, 2009) can be deployed as interoperable tools in the same geospatial infrastructure.

To reach this goal, possible approaches to merge the top-down SDI model with the bottom-up or Volunteered Geographic Information (VGI) geo-infrastructures model (Craglia 2007; Goodchild 2007b, Gould, 2007) are investigated. The key idea behind our methodology proposal is to change the role of SDI users, turning them from pure consumers into active participants playing a more interactive role (Budhathoki et al, 2008) and providing and integrating new resources.

1.3 Contributions

The main contributions of this work are the investigation and proposal of a methodology and a technological solution to help users work more efficiently and participate more actively in GII. Figure 1 illustrates conceptually the components of the ServiceFramework which are the implementation of the technological solution addressing each of the mentioned problems.

First we propose a conceptual architecture and a service design methodology to add processing capability to geospatial infrastructures. With examples, our work shows how exposing scientific tools as standard web processing services has the advantage of eliminating the need for local computer power and maintenance, and provides remote access to reusable data and tools needed for scientific modelling. The major actor of this contribution is the *Process-Wrapper*.

Secondly, we propose components, supported by the mentioned architecture, to assist users in the building and maintenance of these infrastructures by deploying new data resources. We aim to demonstrate how a high multiplicative factor can be achieved in knowledge sharing when hiding the complexity of SDI deployment mechanisms. By providing simpler participation mechanisms, we address the challenge of changing the user's role to become a provider in the development of information infrastructures. In this case, the *DataWrapper* will deal with the provision of these mechanisms.

Finally the ServicePublisher component deals with automatically register the generated services in open Catalogues so it addresses the improvement of the resources visibility.

To accomplish our goal, we have modelled and developed tools to automatically deploy geospatial services based on standard interfaces by wrapping geospatial data and tools provided by users, so that users can easily share their data and tools with the scientific community. This proposed methodology has the added value that scientific data and tools wrapped as standard web services can be reused in other scenarios. Also, this provides for a richer user participation, and therefore, increased accessibility to resources of possible interest to others.

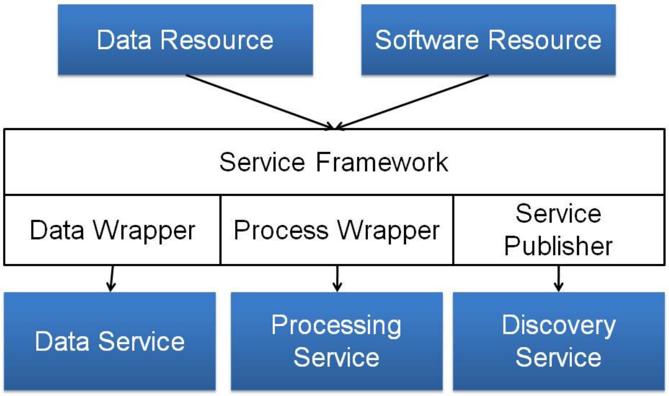

Figure 1. Contributions graphical overview: ServiceFramework conceptual components.

1.4 Context

This section describes the most relevant projects related to our work.

1.4.1 AWARE Project

One of the scenarios to evaluate our proposal has been developed within the context of a 6th Framework Programme European Union Project. AWARE (A tool for monitoring and forecasting Available WAter REsources in mountain environments, see http://www.aware-eu.info for further information) offers geospatial tools to monitor and forecast water resources.

Within this context, we investigate how to help scientists working in distributed GII. We providing them with plug and play tools to offer scientific processing capacities to share and run their models in GII. The fundamental idea consists in wrapping scientific routines as web processing services deployed on SDIs to assist users to share tools and information (Kiehle et al., 2006) (Friis-Christensen et al., 2007).

The scenario presented focuses mostly on the service layer of the AWARE web application. We will provide to the service layer a library of geospatial services that prepare and present all geospatial data necessary for running hydrological models. We describe how these services (implemented following standard specifications) offer the capability to create a distributed application for water resources management, with the added value of being reusable in other application scenarios. Furthermore we demonstrate how we can support GEOSS (Global Earth Observation System of Systems) generating standardized services following the European directive INSPIRE technical approach to fulfil the requirements of maximizing the interoperability.

1.4.2 EuroGEOSS Project

The second scenario to evaluate our proposal is developed within the context of the 7th Framework Programme European project EuroGEOSS (the European approach to GEOSS, see http://www.eurogeoss.eu for further information). EuroGEOSS pursues the improvement and establishment of interconnection among systems and resources as well as the identification of options and interfaces to take benefit from multidisciplinary data and tools available at global, national and regional levels. EuroGEOSS demonstrates the added value to the scientific community and society of making existing systems and applications interoperable and used within the GEOSS and INSPIRE frameworks. EuroGEOSS will demonstrate this in three areas: forestry, drought and biodiversity.

To limit the scope of this dissertation we will work and evaluate our proposal in a forestry scenario. Within the EuroGEOSS GII we will assist users in deploying new resources. The use of the proposed mechanism, the ServiceFramework, by forestry experts, will help to increase the availability and visibility of forestry resources in the EuroGEOSS systems in an interoperable way at global, regional and local levels.

1.5 Structure of thesis

Chapter 1 has introduced the framework of our work, some key issues of availability of geospatial resources and requirements within environmental domains. Furthermore we have seen the difficulties of SDI stakeholders to find specialized tools to perform analysis and to integrate new resources due to a top-down building philosophy of SDI. In addition, we have remarked the general points of our contributions. Chapter 2 starts with the background of GII based on standard web services and summarizes some of the most relevant works related to our work. Chapter 3 describes the general use cases to address and the conceptual architecture designed to be the base of our solution. Chapter 4 describes in detail our proposal following a software development process methodology to analyze, design and implement a proof of concept to assist SDI user to increase the availability and visibility of geospatial resources. Chapter 5 evaluates the solution in two different scenarios within the two European projects framework of this work. Chapter 6 analyzes the conclusions and suggests future work.

2. Related work

This section overviews different interoperability issues, discusses approaches for distributed GII and services, reviews related work on geospatial resource integration and reusability (both processing and data resources). It further describes some of the methodologies used to build GII. To finalize, we analyse some relevant environmental applications and tools related to our scenarios.

Geosciences research is a multidisciplinary field that demands not only heterogeneous data and models but also includes a multitude of expert profiles such as technologists, remote sensing specialists, and geoscientists. These experts, collect, store, manage, organize, and process data using environmental models to produce meaningful information for decision-makers. Moreover, end-users increasingly participate in the generation of geospatial resources. The necessary ingredients for a successful collaborative and multidisciplinary research in this scenario are: (1) new architectural styles which oppose centralized, isolated solutions, (2) the support of distributed processing capabilities, and (3) user knowledge integration.

2.1 Interoperability issues

Since many definitions and levels of interoperability exist, this section identifies the pertinent aspects of interoperability within the context of our work. The following scopes initially need to be considered:

- One of the key concepts of geospatial systems in our context is SDI. Current trends in multilevel SDI development enable end-users to share spatial data in decentralized structures where a top-down structure aims to achieve interoperability while the bottom-up structure aims to integrate user knowledge. So we face the challenge of building geospatial systems based on a common standard in consideration of the heterogeneity of users and their resources (Masser, 2005).

- Our main goals are, on one hand, to add processing capabilities to SDI to make a useful tool for scientific work, and on the other hand, to guide users to participate in the building and maintenance of these systems.

2.1.1 Scope

Several types of interoperability should be considered when discussing interoperability between the information systems. The three interoperability types proposed by the European Interoperability Framework (EIF) are good examples (EC, 2004):

- Technical incompatibilities (e.g. between the technical infrastructures of public administrations from different countries),
- Semantic incompatibilities (e.g. different meaning and usage of documents or information), and
- Organizational incompatibilities (e.g. between different business processes or their goals).

To limit the scope of our work in this aspect, we will restrict the interoperability type to ”technical incompatibilities.” Therefore, we do not deal with semantic or organizational interoperability.

In restricting the type, we have to limit other aspects as well. When we talk about interoperability, we mean to match the heterogeneity of two or more parts so they can interact with each other. To interact successfully, it is important to know the scenario and use cases that will define the real tasks to perform, and consequently, the requirements to achieve. Hence, this will provide the necessary interoperability aspects.

Since our working framework is GII based on the key unit of SDI, we are limited to the use cases in a SDI where we add the new provider role to users. This means that generally the use cases performed in our scenario are to discover, access, visualize, download, process and add geospatial resources.

2.1.2 Interoperability levels in information systems

Related to the technical interoperability type, we can distinguish different interoperability levels. (Bishr, 1998) described six levels between two or more distributed geospatial systems: 1) network protocols, 2) hardware and operative systems, 3) spatial data files, 4) DBMS, 5) data models and 6) application semantics.

Our scenario is a distributed environment, where there are already well-proven standard internet protocols that technically assure us interoperability as we can access remote internet resources by using form instance HTTP protocol. Regarding data file formats, we agree to retrieve them by means of web services that access databases, thereby keeping and hiding the data heterogeneity at the source. Similarly, we access process capabilities by means of web processing services which hide the technical features of the process like programming language, operative system, etc.

In (Sheth, 1999) and (Goodchild et al., 1999), the interoperability levels were reduced to: system, syntax, structure or schema and semantic. Where system interoperability is reached by accepting:

- the internet as a standard between systems
- evolution of II and middleware that support distributed computing by means of Web Services and XML-based standards interfaces.

Syntactic interoperability refers to integrating the elements in various systems such as data formats and standards. It includes the ability to deal with formatting and data exchange supported by standards such as XML and service interfaces such as OWS, WSDL, and SOAP. Thus, we adopt *ad hoc* standards (Sheth, 1999; Feng, 2003) to achieve it. Schematic interoperability is described by common classifications and hierarchical structures while semantic interoperability harmonizes meanings of terms. They can be improved by using metadata standards, data schemas and ontologies (Bishr, 1998).

In our scenario we deal with heterogeneous geospatial resources (data and tools) integrated in II by means of standardized web services. We want to guide users to integrate new resources (in diverse formats) and keep the interoperability of the II. We achieve this by integrating these new resources as standard web services, thus improving interoperability by increasing the standardization and interoperability at the syntactical level. The next sections describe the standards chosen according to international initiatives and directives.

2.1.3 Interoperability in geospatial information infrastructures

One of the first studies that aimed to characterize GIS interoperability was (Bishr, 1998) who demonstrated the need for working on data scattered over services (Abdalla et al., 2007). We have limited the working framework to distributed GII where the key to share resources is the concept of web services (Bernard et al. 2005a). We focus on technological interoperability and concretely at the syntactical interoperability level by adopting components standardization.

There are a wide range of interoperability standards available for the integration of information systems (Mykkäne et al., 2008). Within the geospatial domain, the interoperability is ensured by efforts most prominently by ISO/TC 211 and Open Geospatial Consortium[ii] (OGC) (Bernard et al., 2003).

Interoperability is defined by the OGC as being the ability to: 1) freely exchange all kinds of spatial information about the Earth and about objects and phenomena; and 2) cooperatively, over networks, run software capable of manipulating such information (Buehler and McKee, 1996). In other words interoperability is the ability to exchange and manipulate geospatial resources across distributed systems (Bishr, 1998) without having to consider the heterogeneous format of the source (Phillips, 1999; Masser, 2005).

The OGC has proposed a number of standards with the intention of promoting syntactic interoperability through the use of services (Percivall, 2008). The existing specifications have been proven to help in setting up SDI interoperating geospatial services to access distributed geospatial data (Bernard et al., 2005). Some examples are: OGC Web Map Service (WMS), OGC Web Feature Service (WFS), and OGC Web Coverage Service (WCS) interface specifications. More recent specifications like OGC Web Processing Service (WPS) provide an interface for accessing processing functionality as distributed web services.

2.1.4 Syntactic interoperability of geospatial services

Figure 2 shows the interoperability stack, where different standards are required to reach different level of interoperability to be able to perform discovery, access, visualization and processing in GII.

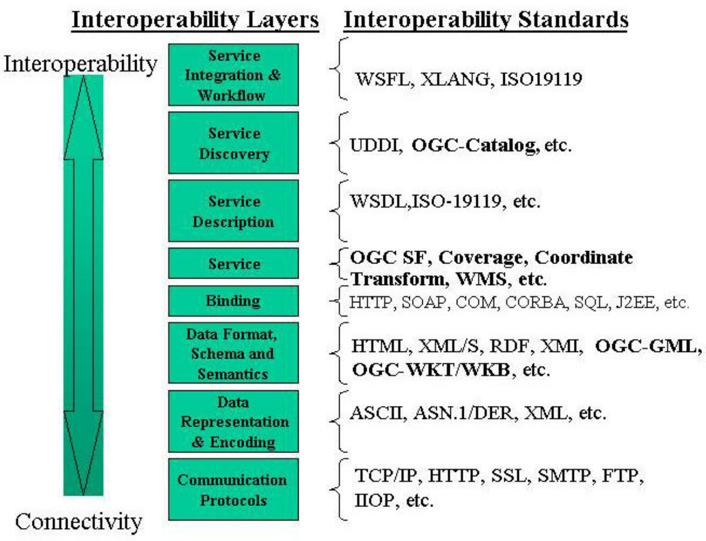

Figure 2. Interoperability and standards stack (extracted from "Standards into Action" (Nebert, 2005)

Since this work aims at integrating resources (both processing and information in SDI), we mostly focus on the service layer of the proposed SOA where we deploy geospatial resources wrapped as web services. The proposed standards to improve interoperability at syntactic level are the OGC specifications, since they provide a standard way to reach the resources in an interoperable way.

2.1.5 Interoperability limitations in geospatial information infrastructures

OGC standards are the *de facto* standards in the geospatial domain. However, due to the dynamic nature of functional requirements, these specifications change in the different versions that appear continuously. This provokes the necessity of changing the technology too often. To solve some of these issues, initiatives like INSPIRE establish a legal framework and technical guidance recommending implementation of rules (IIR)[iii] for the data and services deployed in SDI.

One of the most common problems is that the specifications of OGC standard interfaces are too general, proving a flexibility that is shown in the different specification implementations where each vendor applies its own understanding. Consequently, implementations of the same standard can be incompatible. A possible solution is the conformance test being defined by OGC or the existence of reference implementations.

These, and other difficulties, are observed when achieving interoperability-through-services approach. Issues regarding fault tolerance, server-independent implementation, time-out transactions, privacy, and others show the need for further study (Lacerda and Davis, 2006).

2.1.6 Cross-platform interoperability

Cross-platform interoperability means the ability to share and process information in multidisciplinary environments where different technologies are used.

OGC provides specifications with complementary interfaces (e.g. WSDL, SOAP) to reach synergies with broader-domain web services which follow policies from W3C and other international standardization organizations. An example of this is the collaboration between OGC and W3C to add semantics to the OGC specifications or initiatives to use OGC standards (e.g. OGC WPS) in eScience cyber infrastructures (Lee and Percival, 2008). However, OGC's definition of web services for GI predates the W3C's definition of the Web service architecture (World Wide Web Consortium, 2004) and adjustments are necessary between OGC's and W3C's proposals (Lacerda and Davis, 2006).

2.2 Geospatial information infrastructures (GII)

This section is devoted to describing II as it relates to the geospatial community, and provides an overview of some general concepts and initiatives.

2.2.1 Service Oriented Architectures (SOA) and Spatial Data Infrastructures (SDI)

Scientists and experts need to access data and tools to perform their tasks efficiently. Traditionally, these professionals had limited access to these data and had to collect them from different sources, which is often a highly time-consuming task. The processing of geospatial data, in order to extract useful information, has been done locally by experts using multiple desktop GIS applications. In this sense, GIS applications have been widely used as tools to process input parameters and to produce scientific model outputs used to forecast and assess environmental changes and their impacts on the Earth's resources and hazards.

However, this paradigm – everything locally owned and operated – makes analysis and processing of spatial data tedious and expensive because these applications often involve multiple formats, interfaces and data types. In addition, many GIS applications are underutilized. For example, a software package is purchased only to be able to run one particular routine.

A new trend in providing users with the minimum functionality needed is to deploy geospatial applications under SOA which is based on web services that are effective, simple to use and available in an *ad hoc* manner. These approaches are focused on an architectural style to design applications based on a collection of best practices, principles, interfaces, and patterns related to the central concept of service (Aalst et al., 2007; Papazoglou and Heuvel, 2007). In SOA, services play a key role and become the basic computing unit to support development and composition of larger, more complex services, which in turn can be used to create flexible, *ad hoc* and dynamic applications.

Therefore, a platform for global access to these distributed geospatial resources effectively has to have a technical base founded through services (Rajabifard et al., 2002), deployed on interoperable distributed architectures based on open standards.

The term "service-oriented architecture" refers to systems structured as networks of loosely-coupled, communicating services (Booth et al., 2003). In this way, geospatial tasks now become distributed web services. One of the goals of SOA is to enable interoperability among existing technologies and provide an open and interoperable environment based on reusability and standardized components. In this context the term of "service-oriented science" refers to scientific research enabled by distributed networks of interoperating services (Foster, 2005). (Mineter et al. 2003) pointed out the need for a new generation of environmental applications migrating from the use of standalone programs towards the use of distributed geospatial services using emerging technologies like web services (Alonso et al., 2004) and Grid (Foster et al., 2001), and emphasized the need for modularity and reuse of developing applications as a set of interconnected services.

At the time of implementation SOA-based services must make use of concrete languages and protocols. This is where web service technology gains importance because it increasingly is becoming the choice to implement SOA-based applications. Web services are loosely-coupled independent units, service interfaces describe the functional capabilities and service implementation what a service should execute (Alonso et al., 2004). This principle provides a clean separation of concerns especially between service interfaces (what services offer to the public community) and internal implementations (how services work), thereby promoting one of the goals of SOA: enabling interoperability or the ability of services to interact with minimal knowledge of the underlying structure of other services (Sheth, 1999). Interoperability is achieved (or optimized) by using standard interfaces. Web service technology includes various standards such as Web Service Description Language (WSDL) for the description of service interfaces, Universal Description, Discovery and Integration registry (UDDI) for their advertisement and discovery, and Simple Object Application Protocol (SOAP) that enables communication among services (Curbera et al., 2002).

Related to the general concept of II, many experts believe the terms "information-centric" or "knowledge-centric" would capture the concepts more aptly because the objective is to find and exploit information. Netcentric, or "network-centric" (NEC), refers to participating as a part of a continuously-evolving, complex community of people, devices, information and services interconnected by a communications network to optimize resource management and provide superior information on events and conditions needed to empower decision makers[iv]. NEC offers decisive advantage through the timely provision and exploitation of information and intelligence to enable effective decision making and agile actions (UK MOD, 2005). And it is a concept broadly use in the field of emergency response.

In the geospatial domain, SDI is one of the best approximations to eScience community. The capability of discovering, accessing and sharing a diversity of geospatial resources, among a wide range of actors, is being addressed by interconnected SDI nodes at different scales to build a global GII (Rajabifard et al., 2002; Masser et al., 2007). This net of SDI nodes supports interoperability among multiple geospatial services and large volumes of geospatial resources and client applications to suit mostly user needs in multiple application domains (e.g. Bishop et al., 2000) and forestry (e.g. Davis et al., 2009). Indeed, the use of accurate and up-to-date geospatial information is a crucial aspect in most environmental applications in which, decision-makers must manage and exploit efficiently geospatial resources scattered among numerous agencies and in multiple formats (Mansourian et al., 2005; Rocha et al., 2005; Nayak and Zlatanova, 2008).

Although SDI nodes may rely technologically on cyberinfrastructure to provide increased distributed hardware capacity for handling huge datasets, the SDI paradigm conceptually represents the distributed GIS approach to SOA-based applications in which standardized interfaces are the key to allowing geospatial services to communicate with each other in an interoperable manner, responding to the true needs of users (Alameh, 2003; Foster, 2005; Kiehle et al., 2006; Friis-Christensen et al., 2007).

GIIs are created to facilitate the coordinated production, access, and use of geospatial data among producers and users in an electronic environment (Groot and McLaughlin, 2000; Masser, 2005). SDI plays a key role in supporting users and providers for decision-making where they can discover, visualize, and evaluate geospatial data at regional, national and global levels (Nebert 2004; Masser 2005). SDI can be viewed as an infrastructure linking people to data through the linking of data users and providers on the basis of the common goal of data sharing. (Rajabifard et al., 2005).

Ramamurthy (2006) highlights how cyberinfrastructures evolved from proprietary centralized data systems to open, distributed and standards-based data services that facilitate data integration and greater interoperability.

One of the trends in collaborative science on the Web is the concept of Web Science (Berners-Lee et al., 2006; Shneiderman, 2007). This term covers many aspects in the Web context such as tools, data representation, infrastructures, mechanisms and so on to eventually facilitate discovery, integration, processing, and analysis of data sets from disparate and distributed data sources.

Multidisciplinary research teams require the support of suitable cyberinfrastructures (or eScience) that lets them share distributed data and computing capabilities to achieve their objectives more efficiently (Hey and Trefethen, 2005). This approach has been identified as a solution to the challenge of generic interoperability and data integration in the environmental domain (Denzer, 2005). Cyberinfrastructure allows research teams to share distributed resources (e.g. data sets, processing power) through high-speed networks. Other authors (Denzer, 2005; Goodall et al., 2008) propose cyberinfrastructures and distributed infrastructures as solutions to the challenge of generic interoperability and integration. Several attempts have been made to provide these services in diverse disciplines, such as the Geosciences Network (GEON[v]) project focused on developing a cyberinfrastructure for integrative geosciences research. Another example is the GIIDA[vi] (Gestione Integrata e Interoperativa dei Dati Ambientali) project started in 2008, releasing its System of Systems architecture. One of the main GIIDA objectives is the interoperability with international initiatives for geoinformation. As an inter-departmental project, it aims to design and develop a multidisciplinary e-infrastructure (cyberinfrastructure) for the management, processing, and evaluation of Earth and Environmental resources (i.e. data, services, model and sensors) (Nativi et al., 2010)

2.2.2 International initiatives

In this section we outline some of the most important initiatives devoted to establishing GII to manage geospatial resources to build interoperable systems.

2.2.2.1 Global Earth Observing System of Systems (GEOSS)

The purpose of the political initiative Global Earth Observing System of Systems (GEOSS) is to achieve comprehensive and coordinated observations of the Earth system to improve monitoring of the state of the Earth, increase understanding of Earth processes, and enhance prediction of the behaviour of the Earth system. By expressing interface interoperability specifications as standard service definitions, GEOSS system interfaces assure scalable interoperability. The GEOSS architecture will be based on existing (and new) components. GIS private and public organizations, institutions, research groups and, in short, all GEOSS's participating members ultimately own, manage, and operate the systems making up the GEOSS network of systems. GEOSS's efforts are aimed at areas of great societal impact. The figure below shows the themes in the scope of this initiative.

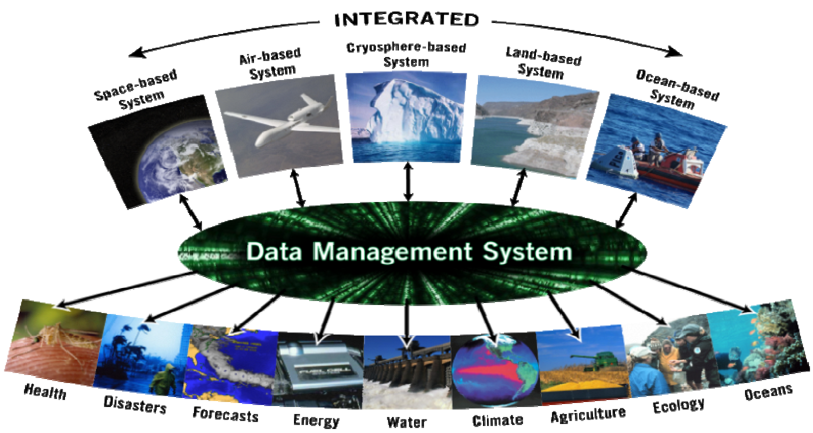

Figure 3. GEOSS scope (GEOSS, 2007)

GEOSS strives to coordinate the interrelationships between the individual systems around the world, while supporting new collaborative initiatives and the development of new systems. Therefore the addressing of global problems must involve international coordination and collaboration to avoid duplication of projects while leveraging synergies between projects to ensure sustained socio-economic development and environmental benefits for society. GEOSS's mission is to ensure that there is no lack of coordination between geospatial and EO systems belonging to different countries so that the mistakes of the past in the detection of natural disasters can be avoided.

In order to meet the above challenges, it is necessary to establish a common framework within which the various systems can communicate and share resources in an interoperable manner, while remaining agile and flexible to allow for changes and the incorporation of new individual systems to the worldwide network. This interoperable framework defines GEOSS's common architecture which promotes the use of common principles, rules, techniques, and standards for all GEOSS systems (GEOSS, 2008).

2.2.2.2 Open GeoSpatial Consortium (OGC)

GEOSS relies on existing GII like SDI nodes as institutional and technical precedents. In GII interoperability is pursued by means of standardized services mostly implementing OGC interfaces. The OGC is an international consortium of companies, government agencies and universities participating in a consensus process to develop publicly available standards specifications. It is the major industry standards organization in the GIS community.

2.2.3 European Initiatives

In Europe there are a number of initiatives to create a technical, political and social infrastructure to support an interoperable space for sharing and managing environmental information and tools. This section presents two of these initiatives which aim to promote the development of GII and EO systems.

2.2.3.1 GMES (Global Monitoring for Environment and Security)

Global Monitoring for Environment and Security (GMES) is a European initiative (Directorate General Enterprise) for the implementation of value-added information services dealing with the environment and security. GMES applications are assumed to be based on observation data received from EO satellites and ground-based (*in situ*) information.

GMES is the European solution responding to the needs of citizens in Europe to access reliable information on the status and evolution of their environment and to ensure an improved security. The purpose of GMES is to deliver Information services based on EO data. GMES is the European participation in the worldwide monitoring and management of planet Earth and the European contribution to GEOSS (Díaz et al., 2009b).

2.2.3.2 Infrastructure for Spatial Information in Europe (INSPIRE)

Adopted as a European directive in February 2007, INSPIRE (Infrastructure for Spatial Information in Europe) (INSPIRE, 2007) sets out a legal framework for the European SDI, with regard to policies and activities having environmental impact. INSPIRE is actually based on GIIs which have already been set up and are managed by each member state, thereby creating an infrastructure of SDI nodes that are operational at a national, sub-national and thematic level for sharing and access to data in multidisciplinary and cross-border projects.

SDI initiatives as a whole contribute to GEOSS by providing a portfolio of standards, protocols, and interfaces to allow geospatial data to be accessed and exchanged. This set of specifications and standards promoted by INSPIRE considerably enhances interoperability between the services and components provided by SDI nodes. In short, INSPIRE is made up of SDI nodes (member states and autonomous community, regional and local governments) managed independently but under the principles of collaboration and reuse.

The purpose of such an infrastructure is, in the first place, to support the formulation, implementation, monitoring, and evaluation of community environmental policies, and to overcome major barriers affecting the availability and accessibility of pertinent data.

The INSPIRE Directive addresses the need for web services to discover, view, transform, invoke, and download geospatial data which enable stakeholders to share data in the multilevel hierarchy (INSPIRE, 2007). Such web services require technical specifications commonly agreed upon by the Member States for the interoperability and harmonization of their SDIs (INSPIRE, 2007). Currently, INSPIRE recommends the OGC specifications and existing OGC Web Services (OWS) standards as technical guidance for implementing GI web services (INSPIRE, 2008).

The INSPIRE technical architecture includes metadata, spatial data sets, and network services within a layered architecture that differentiates the Presentation layer (applications and Geoportals), the Service layer, and the Data Sources layer, as illustrated in Figure 4. Essentially, client applications access geospatial data stored in repositories through services in the middleware layer.

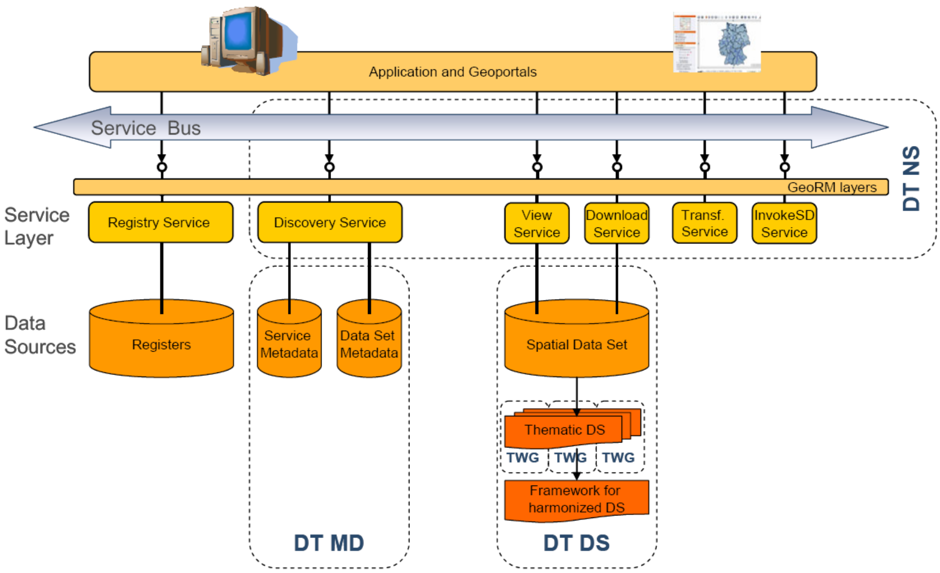

Figure 4. INSPIRE technical architecture (INSPIRE, 2007)

2.3 Geospatial services

Many of the benefits of general services can be extrapolated to geospatial services as well. Services are basic pieces that allow users to access and share information faster and more efficiently by essentially decoupling service description from implementation. What makes geospatial services slightly different from "common" services are the inherent characteristics of geospatial resources on which they operate (Granell et al., 2007). There are a great variety of existing data models, data formats, data semantics, and spatial relationships (contains, cross, touch, etc.) that are, in practice, the limiting factors to ensuring true geospatial interoperability (Ramamurthy, 2006). Nevertheless, service-oriented applications involving geospatial data are still possible. These applications increase their interoperability in part because the geospatial community, under the auspices of the OGC, has proposed specific interface descriptions. Some of these interfaces are complementary to those used for web services (e.g. WSDL, SOAP) while others are more appropriate for dealing with the "special" features of geospatial data (e.g. offering better support in defining geospatial data schemas). That is to say, SOA and web services principles remain intact but the main difference resides in the description languages used.

The need to interface heterogeneous legacy software and data resources can be overcome by wrapping them as web services in a SOA (Ramamurthy, 2006) which lead us to an increment in the interoperability between these components.

Ramamurthy describes how, as a result of the aforementioned trends, the last decade has seen an evolution of data systems like EOSDIS (Earth Observing System Data and Information System) towards a more layered and open architecture, while new data systems have been built and deployed using many open source and standards-based technologies, e.g. the National Oceanic and Atmospheric Administration (NOAA) National Operational Model Archive and Distribution System (NOMADS) (Rutledge et al., 2002), Community Data Portal (Middleton, 2001), Earth System Grid (Foster et al., 2002), data system at the British Atmospheric Data Centre (Lawrence, 2003), and the U.S. Integrated EO System, which implements GEOSS services within a web-enabled component-based architecture, to maximize the value of EO resources (Hood, 2005;IWGEO, 2005).

Regarding service composition (discussed in chapter 5 when evaluating the proposed solution to add processing capabilities to SDI), we have designed processes with different levels of granularity. The coarser-grained processing services are implemented as a chain of other thinner-grained services, act as orchestration modules (i.e. they call to other processing services available in the GII). As defined by ISO standard 19119 (2005), the kind of chaining performed in this case is called opaque chaining, where an aggregated service carries out the chain.

Semantic issues also have been widely researched in the web service domain (McIlraith et al., 2001). Several research works have proposed ontology-based approaches to enhance resource discovery and service interoperability in the geospatial domain (Reitsma and Albrecht, 2005; Lacasta et al., 2007; Lutz, 2007; Smits and Friis-Christensen, 2007; Yue et al., 2007), though discovering semantically suitable geospatial services still remains a very challenging task (Lutz, 2007). Semantic aspects are out of the scope of this Ph.D. dissertation.

The IIR proposes a network of services classified in groups according to functionality (i.e. what the service does in terms of capabilities, to embrace all needed geospatial functionalities). Each group is called a *Service Type*. Figure 4 shows the Service layer which contains the INSPIRE Service Types (yellow boxes) as contemplated in the directive. These service types are: Registry, Discovery, View, Download, Transformation and Invoke. Transformation services and invoke services limit their functionality to schema and coordinate transformation, while certain advanced aspects such as service chaining need further discussion and consensus.

Most of the web services deployed in SDIs use interfaces defined by the OGC during the OWS Web Services specifications initiatives[vii], such as those described by Anderson and Moreno-Sanchez (2003) and Caldeweyher et al.

(2006) which have successfully applied the basic services such as Web Mapping Service (WMS) (Beaujardiere, 2004), Web Feature Service (WFS) (Vretanos, 2005) or Catalogue Service for Web (CSW) (Nebert and Whiteside, 2004). However, these services are insufficient to suit the processing and modelling requirements expected in geospatial infrastructures for scientists to perform analysis and generate new information.

Recent works have highlighted the need to incorporate geoprocessing capabilities in distributed applications, leading to so called geoprocessing services. The ability to not only access and visualize geospatial data but also process them seems to be a great benefit for SDI, since this opens the door to creating richer services that might be applied to wider scenarios. For this purpose, a more recent OGC specification, the OGC WPS specification (Schut, 2007), provides interfaces for accessing more complex services and also for wrapping existing off-line processes as web services.

Although in theory implementing services as XML-based web services should increase chances of distributed system interoperability, still many interoperability problems often arise in practice when different tools from different providers are pieced together (Lu et al., 2007; Díaz et al., 2008b).

2.3.1 OGC Web Data Service specifications

Since INSPIRE Implementing Rules are, as much as possible, in conformance with European and international standards, they recommend the use of wide spread standards like OGC and ISO as we see next. Service interoperability is achieved by utilising standard interfaces. Interfaces are critical because they indicate how to interact with available services in a uniform and unambiguous manner. It is crucial that descriptions for service interfaces are widely published and become standards for widespread use.

Most OGC specifications and standards are devoted for spatial data discovery, abstraction, access, and integration while others are devoted to process and transformation. The OGC Service tier is illustrated in Figure 5.

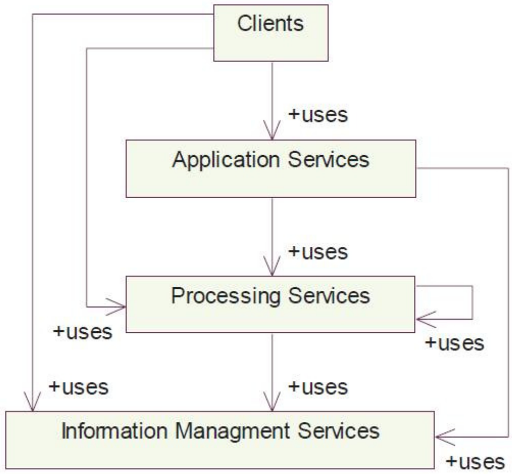

Figure 5. OGC Service tiers (OGC Reference Model)

The specifications regarding data or application services relevant in this context are:

The OGC Web Mapping Service (WMS, de La Beaujardiere, 2004) is the standard *de facto* when adding map visualization capabilities to an SDI. The IIR draft of INSPIRE View Service type recommends the use of this specification when deploying INSPIRE compliant applications.

The OGC Web Feature Service (WFS, Vretanos, 2002) defines interfaces for data access and manipulation operations on geographic features using HTTP as the transport protocol. Via these interfaces, a web user or service can combine, use and manage spatial data – the feature information behind a map image – from different data sources by invoking WFS operations on geographic features and elements such as creating a new feature instance, deleting, updating, etc.

The OGC Web Coverage Service (WCS, Whiteside and Evans, 2008) defines a standard in interface and operations that enables interoperable access to geospatial "coverages." The term "grid coverages" typically refers to content such as satellite imagery, digital aerial photos, digital elevation data, and other phenomena represented by values at each measurement point.

2.3.2 OGC Web Data Service implementations

There exists many implementations of the interfaces OGC WMS, OGC WFS and OGC WCS, including both privative and open source software. To reduce the candidates, we have studied the open source implementations. Sanz and Montesinos (2009) overviewed some of the existing open source implementations of OGC specifications, where, among others, they describe Mapserver[viii], Geoserver[ix] and Deegree[x] as the most common frameworks offering open source implementations of these OGC specifications. To implement the proof of concept, Geoserver is the implementation of choice. Geoserver offers the implementation of the three interfaces and its latest version provides an API called Geoserver RESTful API[xi] which permits us to deploy resources and configure services programmatically.

2.3.3 OGC Web Processing Service specification

The OGC WPS specification (Schut, 2007) provides access to calculations or models which operate on spatially referenced data. The required data can be available locally or delivered across a network. The calculation can be as simple as subtracting one set of spatially referenced numbers from another, or as complicated as a global climate change model.

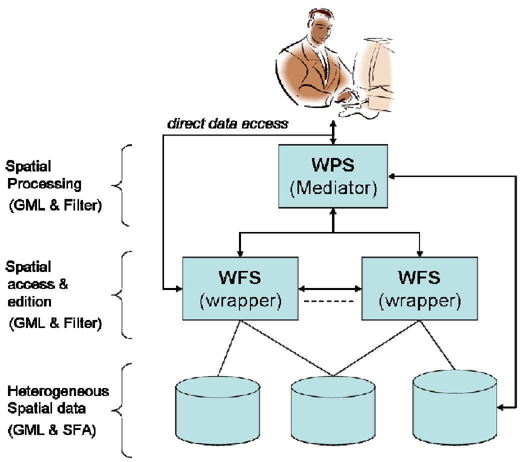

Figure 6. Data Integration and Processing OGC Services

Figure 6 shows how WPS services may play the role of mediators that provide an integrated interface to different integration services (wrappers).

The WPS describes a common interface for services offering processing operations on spatial (vector and raster) and non-spatial data. WPS provides mechanisms to identify the data required by the process, initiate the process, and manage the output so that it can be accessed by the client.

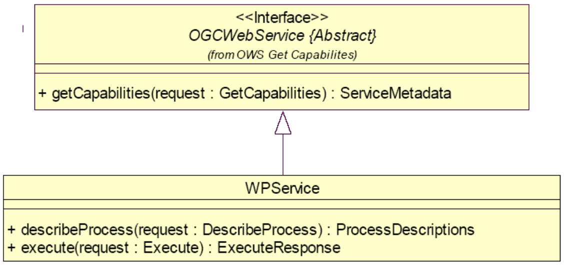

Figure 7. WPS interface UML diagram

As we can see in Figure 7, OGC WPS interface provides three methods: *getCapabilities* method (common in other OWS services) identifies the processes offered and the specific capabilities of a Processing Service instance. By invoking the *describeProcess* method, we request and receive detailed information about one or more processing operations that can be executed by an *execute* operation, including the input and output parameters.

The basic operational unit of the OGC WPS is the notion of process – a geospatial operation, with inputs and outputs of a defined type. This means that a given WPS instance may offer one or various processes as normal web services do. Figure 8 shows how a WPS-client communicates with a WPS instance by issuing the three types of requests. A request can be sent to the WPS instance via HTTP GET with parameters provided as Key-Value Pairs (KVP) or via HTTP POST, with parameters supplied in a XML document.

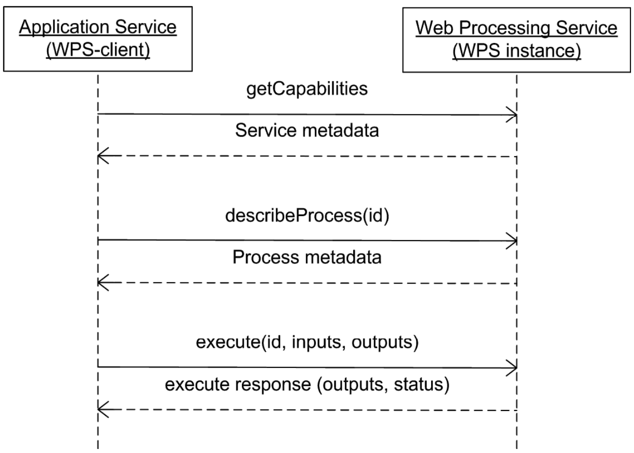

Figure 8. Synchronous interaction between a WPS-compliant client and a WPS service instance

In the present dissertation project, OGC WPS goes beyond providing unique geoprocessing routines because it wraps both spatial and non-spatial processing services, leading to increased interoperability between OWS services and general purpose web services.

A WPS process is normally an atomic function that performs a specific geospatial calculation. Chaining of WPS processes facilitates the creation of repeatable workflows (Schut, 2007). WPS processes can be incorporated into service chains in a number of ways:

- A BPEL engine can be used to orchestrate a service chain that includes one or more WPS processes.
- A WPS process can be designed to call a sequence of web services including other WPS processes, thus acting as the service chaining engine. This is the chaining performed in our use case.

2.3.4 OGC Web Processing Service Implementation

There are several implementations of the OGC WPS specification. Python Web Processing Service (PyWPS[xii]) (Cepický and Becchi, 2007) is an open source python framework that implements the OGC WPS specification version 0.4.0. PyWPS includes native support for GRASS[xiii] GIS (Geographic Supported Analysis Support System), as well as for the R project for Statistical computing (http://www.r-project.org/). The Tigris WPSInt[xiv] implementation is an open source Java plug-in for Spring[xv] – a Java framework for developing web applications – to support the OGC WPS version 0.4. Contrary to PyWPS, the Tigris WPSint implementation has recently added support for SOAP and WSDL. This feature helps to converge SOA-based services and OGC-based services because both kinds of services may be combined to build heterogeneous service chains since both use the same service interface (WSDL). The Degree[xvi] project is an open source Java framework that implements the OGC WPS integrated in their platform WPX, which supports several OGC Specifications.

The 52N Web Processing Service is an open source Java framework developed by the 52 North Open Source Initiative[xvii] that enables the deployment of WPS services. It features a pluggable and extensible architecture for processes and data encodings based on the notions of repositories, which provides dynamic access to the embedded functionality of the WPS already registered in the framework (Foerster, 2006). This has been the implementation of our choice to design and implement the web processing services to be deployed to add processing capabilities on GII.

2.4 Geoprocessing in Spatial Data Infrastructures

Recent works have highlighted the need to incorporate geoprocessing capabilities in distributed applications, leading to so-called geoprocessing services. The ability to not only access and visualize geospatial data but also to process them seems to be a great benefit for SDI. It opens the door to creating richer services that might be applied to wider scenarios. This would provide users with processing tools in the form of reusable standard web services so they could, in a light-weighted distributed way, perform analysis.

Previous to the official stable release of OGC WPS specification, some authors have performed geospatial data processing on SDI like the work described by Scholten et al. (2006) using a SOAP approach. Another work related with distributed geospatial processing that does not use SDI services and standards is the work of Shen et al. (2005) which showed the feasibility of performing image processing with web service technology. However, the authors do not use OGC specifications and interoperability is thus limited.

More recently, Michaelis and Ames (2009) performed a feasibility study of the WPS specification in client-side applications. They conclude that "the WPS proposal was found to be workable as currently designed, and is indeed suitable for many GIS tasks." Kiehle (2006) and Yang et al. (2008) also discussed the use of WPS-based geoprocessing services applied to real world examples. Friis-Christensen et al. (2007) proposed a similar approach for distributed geoprocessing based on SDIs, though they proposed a different approach for creating geospatial services by concentrating all required functionalities in a single, publicly accessible geospatial service. Although their system has advantages in terms of performance, it decreases flexibility and reusability.

Lee and Percival (2008) defined the OGC WPS as a first step of OGC in the direction to provide distributed computing capability in the geospatial and eScience frameworks. We will demonstrate how WPS can be use to generate user information and how we can assist users to integrate and deploy this information back in the information system as a standardized component.

Furthermore, (Scholten et al., 2006) stressed the scarcity of model-based approaches to let researchers analyze and measure system implementation performance. They presented a performance evaluation of an SDI where they concluded that SDI performance can be increased by applying diverse techniques in these four influential performance issues: caching, network adaptation, data granularity, and communication mode.

2.5 Wrapping and sharing geospatial resources as web services

The integration of data resources like heterogeneous database systems is necessary to provide access to data distributed across multiple systems (Abel, 1998). Regarding wrapping of data as web services to achieve interoperability and reusability, Fileto (2001) described several approaches for data integration over the Web, like the common wrapper techniques (Roth and Schwarz, 1997) and standards for exchanging geographical data among systems (Albrecht, 1999). Also, in Díaz et al. (2009a) we found a more recent study in geospatial domain to integrate heterogeneous data sources using OGC specifications.

In relation to processing resources, using wrapping techniques in the field of web applications is not new and many migration strategies have been studied to turn existing software into web services (e.g. Di Lorenzo et al., 2007; Canfora et al., 2008).

In a broader scope outside the geospatial domain, research works like Gonzalez-Escalante et al. (2005) proposed the automatic generation of web services to publish data, hiding the technology and letting the user with no programming skills be exposed to resources in distributed environments for future reuse. The users describe a tool to simplify the query process to distributed databases, the information is interchanged between heterogeneous systems and the tool finally automates the creation of Web Services through a graphical interface.

In the biomedical domain, Krishnan et al. (2006) described Opal as a tool to wrap scientific applications as web services. Similar work is performed by (Li et al. (2008) where they described GSLab, a tool to automatically wrap legacy software as web services so they can be published in Grid environment.

To achieve syntactic interoperability, TSIMMIS (Garcia-Molina et al., 1995) used a mediator approach to combine information from several sources containing textual and semi-structured data. Data sources are encapsulated using wrappers or translators that logically convert the data to a common information model by translating information requests and results to this common model. The mediator layer above the wrappers is responsible for a) routing queries to sources and b)processing the results. The system generates wrappers and mediators automatically for a set of specified rules. TSIMMIS provides a framework for users to specify information integration which may be done manually or in a semi-automated manner.

Many works have described SOA-based architectures to support the sharing of geospatial resources (data and tools). The use of client-server model in web mapping applications or the distributed GIS approach to SOA-based applications represented by the SDI paradigm stimulates the use of standard formats and exchange protocols, and permits the distribution of geospatial functionalities to relevant users (Alameh, 2003; Foster, 2005; Friis-Christensen et al., 2007; Kiehle et al., 2006; Yang et al., 2008; Brunner et al., 2009; Fook et al., 2009, Friis-Christensen et al., 2009; Granell et al., 2010).

Friis-Christensen et al. (2007) described one of the pilots of the Orchestra project (ORCHESTRA, 2008), in which they described an application for forest fire statistics built using distributed services on top of an SDI. However, in this work they do not consider splitting the application functionality as reusable geoprocessing services. Similar ideas and methods have been discussed by Kiehle (2006) in the sense of applying WPS on top of basic SDI resources. Moreno-Sanchez et al. (2007), Kiehle (2006) and Díaz et al. (2008b) proposed similar approaches to run geoprocessing on top in SDI using OGC standards.

Brunner et al. (2009) proposed as future work the implementation of standards like OGC WPS in their system to provide geospatial processing to support collaborative and rapid emergency response. Similarly Brauner and Schäffer (2008) described an approach to expose the functionality of the GRASS software as OGC WPS and calls for the need to generate process descriptions automatically.

Related to wrapping existing off-line algorithms in open source projects to expose them as web processing services, we have mentioned PyWPS (Cepický, 2007). PyWPS allows making native connections to GRASS routines. These routines are encapsuled as contained processes in an OGC WPS.

The GeOnAS project (An Online Analysis System Based on Service Oriented Architecture) (Di et al., 2007) focused on data discovery, data analysis, and data visualization via the Web. It also provided many geospatial processing functions for manipulating and analyzing vector and raster geospatial data by using PyWPS.

Regarding the WPS client interface, we found some works like the uDig desktop client extended with WPS client (Schaeffer and Foerster, 2008) to access different WPS instances. Our WPS API Java library is an independent library implementing different versions of the WPS specification that can be integrated in multiple applications.

We focus on architectural aspects related to integration and reuse of geoprocessing services within SDI contexts (Díaz et al., 2007; Friis-Christensen et al, 2007) rather than merely implementation aspects. We encourage geoprocessing services to include traditional geospatial and statistical functionality (e.g. charting and tables).

Brunner et al. (2009) described a system to provide distributed geospatial processing to support collaborative and rapid emergency response, and also a system for storing results in public databases. They also proposed as future work the implementation of relevant standards as OGC Web Processing Services (WPS) (Schut, 2007).

Rocha et al. (2005) described how MEDSI uses OGC Web Services to provide and manage geospatial information, thereby creating crisis centres with geospatial services accessible in and out of the centres. Abdalla et al. (2007) presented a case study to demonstrate the utility of interoperable web services for disaster management and discussed the strengths and weaknesses of leveraging GIS interoperability.

Fook et al. (2009) presented a Geoweb service architecture that supports the sharing of modelling results and enables researchers to perform new modelling experiments. They presented the Web Biodiversity Collaborative Modelling Services (WBCMS). The Access Processor context presented in their work supports queries and displays model instances. Besides WMS and WFS, the Access Processor can access two special services, WMIQS (Web Model Instance Query Service) and WMIRS (Web Model Retrieval Service), to handle the queries and retrieve the necessary data, respectively. The Model Processor enables researchers to build new models and visualize model instances. In the developed prototype within the OpenModeller Project, users are allowed to add metadata and reuse the results of the models. Although they have used OGC interfaces for web data services, they do not mention how processing capabilities can be accessed from outside their architecture. In addition, they proposed to reuse processing results by storing them as files described in Catalogues Services.

Using the 52 North WPS framework as standard open source tools allows for developing more sophisticated processes (Foerster et al., 2010). They implemented and deployed generalization processes in open architectures, sending the results of the processes to be served by the OGC WMS implementation (developed by Geoserver). They concluded that these results can be accessed from any WMS client and increase syntactical interoperability. The use of profiles in the WPS specification helps reach semantic interoperability.

2.6 User generated information

With the emergence of Web 2.0, ordinary citizens have begun to produce and share GI on the Internet. These Web 2.0-based geospatial activities show that users are willing to engage more actively in the production and supply of information. This gives rise to a new phenomenon which has been variously named ‘neogeography’ (Turner, 2006), ‘cybercartography’ (Tulloch, 2007), or ‘voluntary geographic information’ (VGI) (Goodchild, 2007b).

The increasing popularity of the Web 2.0 philosophy has brought with it new paradigms of design: applications with a high degree of interaction and multimedia effects, content generation by the user, distributed information, importance of aesthetic value, etc. It seems obvious that there is a need for geospatial information systems to join the current state of the Internet where more attractive tools build more appropriate mechanisms for user collaboration and user-driven data management, allowing for improved maintenance of information systems.

Among others, a good example is Geocommons[xviii], a site that permits users to create their own maps and share them with the rest of the community (insert reference). One of the success factors is the ease of use in generating, modifying and deploying the content. Users do not need to be in touch with the technology underneath to add new resources in the GII. But these resources can be shared only within Geocommons applications.

An example of intent to assist users to share resources by means of standard SDI services is the OpenGeo suite[xix]. This suite offers the user a toolbox to deploy and configure, among others, services implementing the OGC specifications.

There are many authors that address questions concerning the increasing number of people participating in VGI while SDIs traditionally face problems to attract users. While VGI participants freely contribute GI, stakeholders in SDIs are often reluctant to share information. In this context to enable SDIs to accommodate VGI and derive utility from their synergy, Budhathoki et al. (2008) proposed to reconceptualize the notion of the SDI user from a passive recipient to an active information actor. Budhathoki et al. (2008), in their reconceptualization of the user role, described how users have to appropriate innovation which occurs at several levels of increasing sophistication: reinterpretation, adaptation, and reinvention.

At the administration level, Masser et al. (2007), presented some challenges to implement SDI to spatially enable government, like the need for new and more inclusive models of governance to enable stakeholders to participate in SDI implementation. And the need of data sharing on a massive scale for SDIs to become fully operational. Here we point out the need of hiding complexity and technology to permit user participation and increase the availability of resources in SDI.

SDI researchers have called for a user-driven SDI model (Williamson, 2003; Masser, 2005; Budhathoki and Nedovic-Budic, 2007) which relates to the hybrid SDI that incorporates VGI.

Combining scientific knowledge and public information is not new, according to Jankowski (2009). Dienel (1989) developed the ‘‘citizen panels’’ in the

1970's involving experts and citizens to make everybody participate. In the context of municipal activities, Carrera and Ferreira (2007) also proposed to capture and utilize the 'city knowledge' from those close to a particular phenomenon with the richest geospatial knowledge. Another example is the SDI being implemented to manage natural resources in the Amazon area, where because of EO systems limitations in such a wild environment, there is a need for user participation to integrate their local knowledge (Fonseca et al., 2009). A synergy between SDI and VGI can lead us to an hybrid methodology in building SDI where top-down official approaches meet the bottom-up or user-driven approach.

To reach this goal, we will investigate possible approaches to merge the top-down SDI model with the bottom-up or VGI geo-infrastructures model (Craglia 2007; Goodchild 2007b, Gould, 2007). The key idea behind our proposal is to add the "provider role" to SDI users, turning them from pure consumers into active participants (Budhathoki et al., 2008) where they provide and integrate new resources.

Having a bi-directional SDI where everybody can potentially be a provider can also be the source of a measure of quality of service. Client feedback can be used to establish perceived quality and to identify the quality parameters most valued by a user group (Lacerda and Davis, 2006).

2.7 Environmental applications

There exists an extensive literature about environmental systems where distributed applications deployed on top of SDI are more and more the most common ones. Next we describe some of the works we found relevant to our work because of the theme (hydrological or forestry models and applications) or other common requirements.

Many applications and tools currently exist to enhance the interaction with environmental models, and these possess a varying degree of sophistication and functionality. Most are built on top of well-known geospatial software packages, meaning that for the most part they remain standalone desktop applications (Mineter et al., 2003; Jeong et al., 2006;Pecar-Ilic and Zuric, 2006; Best et al., 2007; Teng et al., 2008;). In contrast to these applications, we find distributed, web-based solutions, normally integrating a web mapping viewer clients which allow the user to visualize multiple datasets (Soh et al., 2006; Goodall et al., 2008), either taken from static repositories or (rarely) as a result of applying data transformations on-the-fly.

Regarding desktop solutions, Best et al. (2007) described a system using the ESRI Model Builder[xx] with which basic OGC services like WMS and WFS are integrated. Basically, processing tasks are embedded in the system, and as such, they are neither widely available for other users nor general enough to be reused in other scenarios. Interestingly, Best et al. (2007) introduced the concept of scientific workflows using geospatial web services in an ecology use case. Teng et al. (2008) presented a tool to support spatially-distributed hydrological modelling built using ArcGIS[xxi]. Though not service-based, this scientific tool, hides the complexity of the computation algorithms behind a user-friendly interface using a stepwise web application.

Pecar-Ilic and Zuric (2006) presented a tool based on Autodesk MapGuide Viewer[xxii] that aims to provide data conversion and transformation operations among different reference systems for Danube River data. Similarly, Jeong et al. (2006) described a hydrology application based on the Interactive Data Language (IDL[xxiii]) software to analyze and visualize hydrologic data. Nevertheless, all of the application examples seen so far follow an "extension" approach, in which existing GIS software packages are "extended" locally to process and display specific datasets.

In the category of distributed applications, Soh et al. (2006) described a web application to identify drought-vulnerable regions. They proposed a combination of data mining techniques to characterize the behaviour of water basins and classify them according to the drought index. It is important to note that geoprocessing capabilities are not present in terms of distributed geospatial services accessible via Web protocols (Soh et al. 2006).

In relation with distributed environmental application, we found SDI deployments. For instance, in emergency and risk domain, Scholten et al. (2008) described the SDI for emergency response in Netherlands.

Bayarri and Capo (2010) described the impact of the Andean Information System for Disaster Prevention and Relief (SIAPAD) as the most extensive SDI implementation in South America, and a pioneer example in the area of disaster risk management. As a result, more than 5,000 information products are now accessible through the GEORiesgo portals covering all risk management processes. The use of standards for Web services now allows institutions to publish the results of their activities and projects to a wide audience, increasing their visibility and social recognition. A worrisome finding of the project was how often the same work is replicated in parallel initiatives and projects which do not share or build upon each other. Another critical topic, since it greatly affects the sustainability of the system, is the need of continuous support to the participating institutions.

Concrete examples of web service technology applied to environmental models, and specifically to hydrology and forestry, are actually very limited. Goodall et al. (2008) explored to some extent web service interfaces to provide data access for the National Water Information System in the United States. Nevertheless, none of the previously mentioned applications provide distributed processing capabilities when executing on-line environmental models. In the following chapters we describe the conceptual architecture of our system and the strategy followed to design, compose, and reuse services that permit the re-organization of environmental applications as a distributed network of interoperating services.

Although not critical, we have implemented the software component using open source software in our use cases. Some authors support that the use of open source software and open standards for publication of geospatial datasets in fire information systems has in fact proved to be beneficial (Friis-Christensen et al.,2006; McFerren et al., 2009; Giovando et al., 2010).

Because of the impact of fires at the global level, it is important to identify fire risk and provide early warning systems at different geographic scales (de Groot et al., 2006). Several fire information systems exist from local to global scale level (e.g. EFFIS[xxiv] (European Forest Fire Information System) in Europe, CWFIS[xxv] (Canadian Wildland Fire Information System) in Canada, AFIS[xxvi] (Advanced Fire Information System) in South Africa, USFS GeoMac[xxvii] in the United States, and FIRMS[xxviii] (Fire Information for Resource Management System) at the global scale. A common challenge for these systems is to provide critical geospatial data in a quick and reliable way before and during fire emergencies. It has been demonstrated that geospatial support systems designed and implemented with interoperability standards provide easier access to this type of critical information (Friis-Christensen et al., 2006).

Giovando et al. (2010) described how in EFFIS, layers (raw and processed data) are presented into the map viewer through an internal WMS connection. Furthermore, public standard services like WMS, WFS, and WCS will be available for most layers starting in 2010.

At local levels within Spain, the Andalusia regional government (Junta de Andalucía) has developed a system named SIGDIF (de Sarriá et al., 2007) to manage forest fires emergencies. The SIGDIF system contains a couple of subsystems. The first performs simulation tasks for forests fires based on different data layers. The second handles the availability of the forest fires units and human resources involved in the fire. In the same context the SIGIF (López and Poyatos, 2007) from the Valencian regional government (Generalitat Valenciana) focuses on the prevention and surveillance of forest fires. This system allows a decentralized access to the spatial data, including fire simulation and location of the forest fires.

At global and local levels, EFFIS, SIGDIF and SIGIF describe the use of OGC standards to publish spatial data, though most of the time they reduce the use to the intranet environment.

2.8 Summary

The general situation of these GII is that they do not allow collaborative participation of users; the GIIs are not available to external users to add or share new user generated information. Moreover, they do not expose their processing capacity (model or related process) as accessible web services implementing, or not, the OGC WPS specification. Thus, their reusability in other scenarios is very limited.

In relation to geoprocessing, we propose how to improve the availability of geoprocessing resources by describing a methodology to design and implement processing services maximizing their interoperability and reusability. We have designed and developed a library of WPS that can be reused as building blocks to create distributed applications on top of the upcoming European Spatial Data Infrastructure built under the INSPIRE Directive. Our goal is to help scientists in approaching this INSPIRE philosophy to more efficiently meet their requirements.

In Chapter 4 section 3 we describe how to increase the number of reusable scientific tools implemented as software resources in GII. We do not approach automatic methodologies, but we propose design decisions to increase the reusability by varying the level of granularity balanced with efficiency of these resources. Moreover, we propose programming techniques to assist developers in wrapping existing software as OGC WPS to expose this functionality as standard components to help building distributed applications on top of SDI.

In Chapter 4 section 4 we describe how to reuse processing results or other data resources by wrapping them as standard data services using INSPIRE view and download service types. The aim focus on helping the user integrate their generated information into GII to be reused in other scenarios and helping to keep GII up-to-date. We can improve interoperability and reusability by focusing on the implementation of standard interfaces.

3. Architecture

This chapter is devoted to analyzing the requirements of GII, particularly in the environmental domain, and to propose a conceptual architecture that supports the necessary components to fulfil these requirements.

3.1 Introduction

In the literature review section we have described some open issues to address in order for GII permit users to work efficiently:

- Users need to be able to create distributed applications on top of the SDI nodes that form the GII, by reusing standard services available in this GII.
- The GII needs to provide processing capability to generate tailored information according to user needs.
- This GII needs to provide effective access to up-to-date information.
- To maintain this GII, users must participate by deploying new resources.

To address these issues we design a conceptual architecture which has the purpose of supporting distributed environmental applications on top of GII and providing users with other mechanisms and tools to upload resources to increase their availability and therefore the usefulness of the GII.

In order to design a proper architecture, it is important to revise and address the requirements.

3.2 Analysis of requirements

Next we describe the needs of users according to the general use cases related to three different perspectives: the user, the GII and the applications.

3.2.1 Spatial user requirements

The most common general use cases performed in current geoportals show that the main requirements of spatial users are: visualization, ease of use, interoperability and mashups, and modelling and simulations (Gore 1999; Goodchild, 2008). We extract similar user requirements as described in the SDI cookbook (Nebert, 2004): search and discovery, visualization, features selections, download and analysis, and processing. These are in fact similar to the requirements pointed out by GEOSS as seen in Figure 9 where the use cases from users and providers are shown.

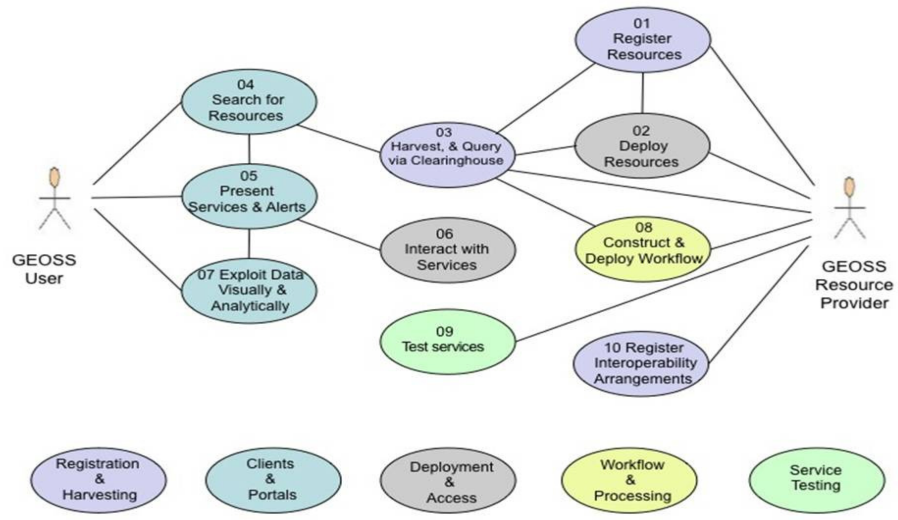

Figure 9. GEOSS technological use cases

In theory, traditional GII support the most common requirements of geospatial users – discovery, access, download and visualization of the data – however, there are still interoperability issues (we will describe these issues in the "Lessons Learned" section of Chapter 6). Note that in this scenario we miss these two points:

- Specialized users such as scientists and decision-makers require advanced services for processing huge volumes of data using specialized models.
- General users are not required/allowed to deploy resources in GII to have a more interactive way to share resources.

These two observations are also mentioned in several related works as we have detailed in Chapter 2.

3.2.2 SDI Building requirements

Two main methodologies exist to build an SDI:

- Bottom-up approach SDI nodes are built through the exchange and consolidation of information from local levels to global levels (Rajabifard et al., 2002). In this hierarchy, lower levels provide detailed information that help in the consolidation of upper levels (Rajabifard and Williamson, 2001; Jacoby et al., 2002; Lacerda and Davis, 2006; Man, 2006).
- In the Top-down approach the global level establishes agreements on standards, procedures and policies to deploy SDI nodes.

Both methodologies would need the establishment of enabling platforms to facilitate access to spatial data and the delivery of data-related services (Masser et al., 2007).

Traditionally deployed SDIs follow the top-down methodology where official providers like public administration deploy resources according to agreed policies and standard components. This scenario responds to the provider-consumer paradigm as shown in Figure 10 where some stakeholders are exclusively providers and others are consumers.

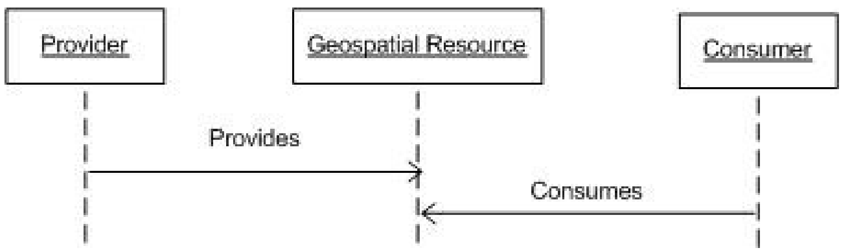

Figure 10. Providers-Consumers paradigm sequence diagram

Bottom-up methodologies where end-users can participate in the SDI building by deploying new resources are currently not performed, although many authors think there is high potential.

In contrast, the Internet and information technologies have evolved to offer mechanisms to share resources to all kinds of users. Whereas the early Web was primarily one-directional, allowing users to view the contents (pure consumers), the new Web 2.0 is bi-directional in which users are able to interact and provide information to central sites (Goodchild, 2007). This means that we have evolved from the pure provider or consumer vision to a more complete role of the user, where the individual can be a consumer and provider (Figure 11). This new paradigm has special interest in the geospatial domain because users generate new resources that do not become integrated into the system.

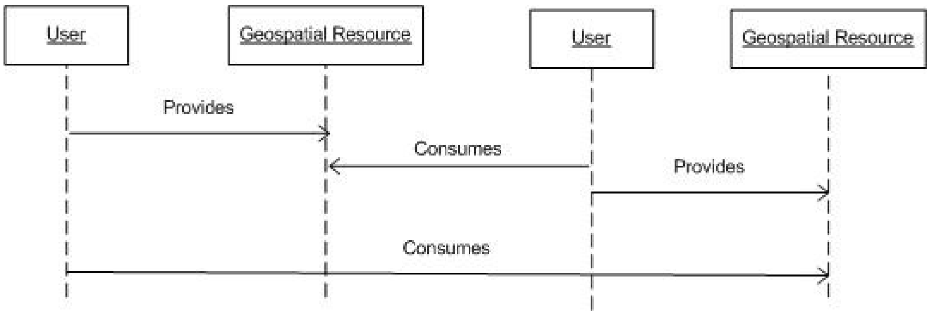

Figure 11. Users as providers-consumers in a bi-directional environment

SDI regularly needs maintenance and refinement due to the dynamic nature of SDI and its resources. The inherent complexity of standardized SDI and the complex mechanisms of deployment get worse as SDI grow (Béjar, et al., 2009a).

At this point, it is important to remark on two issues to be addressed in the development of an SDI:

- Handling the SDI complex deployment mechanisms, and
- Letting end-users become providers as they are involved in resource generation and maintenance.

Hence there are two new requirements:

- Providing mechanisms to guide providers to deploy resources to build and maintain SDI, and
- Applying these mechanisms to providers and to end-users, thus creating a new role for general users who will then become resource providers as well.

This generates a new paradigm in SDI building where top-down and bottom-up methodologies converge. This gives rise to a new hybrid methodology where we witness an improvement in resource deployment and maintenance through user participation, and resource availability through interoperable SDI services would increase considerably as a result.

We add a new use case to expand the functionalities performed by the GEOSS user (shown as a filled red circle in Figure 12). End-users can now deploy new resources in the GII.

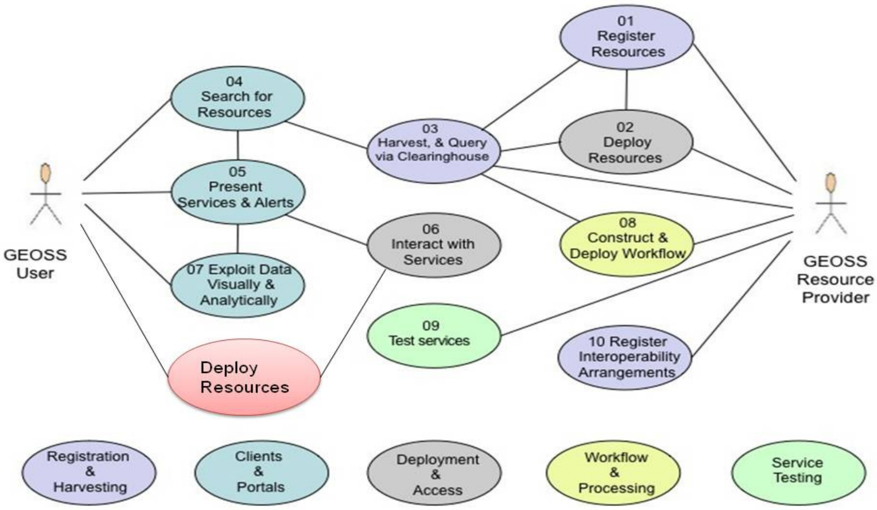

Figure 12. Extended GEOSS technological use cases

3.2.3 Domain applications requirements

Our scenarios are related to the field of environmental sciences, but we would also like to consider the requirement of a particular domain like emergency management because its requirement of rapid availability of up-to-date resources is fully addressed by our work. Figure 13 provides an example of the steps to generate and share multiple resources for decision-makers and actors to work efficiently in an emergency domain scenario. In the first step, a technician accesses data and tools placed in a distributed SDI. In the second step, the technician is assisted to deploy this newly generated resource in a standardized component that is publicly accessible to other SDI users. The third and fourth steps show the added value of this functionality where other SDI stakeholders can rapidly access this new resource and interoperate with it.

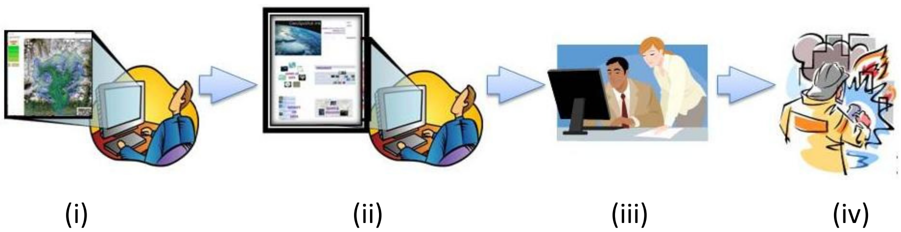

Figure 13. Steps to generate and share information in emergency management: (i) and (ii) resource access, generation and deployment; (iii) and (iv) sharing and collaborative decision-making and actuation

Figure 14 summarizes the steps, we consider relevant, in the resource life cycle. We could determine which of these resource steps are crucial to maximize the availability of resources. Starting with the first circle and going clockwise, resources first must be integrated in the distributed system. We propose to assist users to add resources. The second step consists of the need to publish these new resources in open Catalogues or any other mechanism used in the distributed system to increase the visibility of the resource to the public. The third step illustrates how these resources are discovered and accessed by using the open SDI. In the last step, users process these resources to generate up-to-date information that is then ready to be integrated in the distributed system again. This means that this newly generated resource should persist in a distributed way, rather than being stored locally and becoming isolated from other end-users.

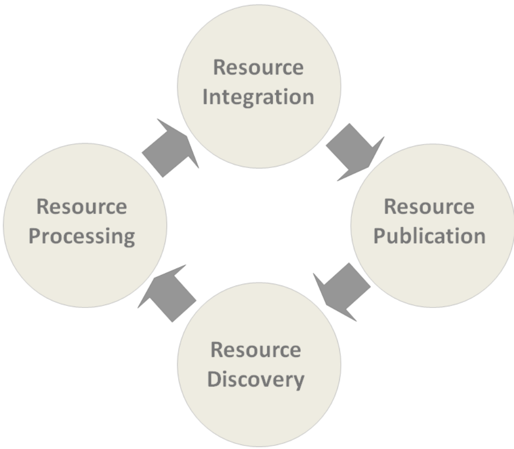

Figure 14. Geospatial resource life cycle in a distributed environment

Geospatial technologies can be used to monitor the environment and to prevent environmental disasters by improving the monitoring of hazards through the development of observation systems, integration of multi-source data and efficient dissemination of knowledge to the concerned people (Nayak and Zlatanova, 2008). This means that in our domain the requirements are:

- To provide and inform decision-makers with tailored output (Brunner et al., 2009). The information should be quickly and easily accessible and presented in an appropriate way to ensure the understanding of the situation (Almer et al., 2008).

- The use of accurate and up-to-date geospatial information to understand the context of the emergency situation (Mansourian et al., 2005; Rocha et al., 2005; Zlatanova et al., 2006).
- Getting the information fast is a key component, and therefore, good availability of the information in the first stages of event is vital for the good management of the situation, thus time is a key issue (Zlatanova, et al. 2006)

The first point has been mentioned in the user requirements section, where the necessity of processing to extract and prepare information is pointed out. Providing the SDI with tools to process and prepare data could help fulfil this requirement. In addition, having data and processes predefined in advance in an SDI will enhance the information delivery (Scholten et al., 2008), which is related to the second and third points.

The second and third points are related to the requirements listed in Section 3.2.1 and 3.2.2. These mechanisms, used to assist stakeholders to deploy resources in an SDI, permit a massive addition of updated resources that would be rapidly available in an SDI in an interoperable way.

3.3 Architecture Principles

The INSPIRE Directive aims to create a Europe-wide SDI supporting cross-scale, cross-language and cross-border interoperability, and accessibility for spatial resources (INSPIRE, 2007). It establishes a legal framework and a technical guidance which recommends deploying spatial resources according to standard protocols and service components.

Based on these principles we propose an architecture based on INSPIRE, and consequently, on its policies and standards to reach interoperability throughout the standardized components. The main contribution of this work is to extend this architecture to address and fulfil the previous requirements:

- Add processing capability to deploy distributed environmental applications that let users perform analysis, extract and prepare information,
- Support a hybrid methodology to build an SDI supporting a user-driven approach to deploy resources, assisting users to deploy geospatial resources to guarantee their updating and accuracy.

To fulfil our requirement in the basis of an interoperable framework and standard components, we have based our conceptual architecture on the INSPIRE technical architecture (see Figure 4).

Our architecture is two-fold. On one hand, it is defined as a service-oriented architecture where the basic unit is the web service, and on the other hand, it is composed of three tiers according to the INSPIRE architecture. Although multi-layered approaches have been proposed in other Web-based GIS applications (e.g. Chang and Park, 2006; Moreno-Sanchez et al., 2007), INSPIRE-based architectures offer more benefits to end-users (and developers) in addition to common technical aspects such as standard interfaces, service types, policies and agreements that globally enhance data and service interoperability.

3.4 Proposed Basic Architecture

Based on the previous requirements and principles, we have designed a conceptual architecture which aims at supporting the deployment of distributed environmental applications on top of an SDI (Figure 15).

The goal of the architecture is to fulfil the general spatial user requirements of search and discovery, access, visualization and analysis of information in a distributed way. The fact that the architecture is based on INSPIRE guarantees that this functionality is provided by means of the standard INSPIRE services deployed on it.

In addition to these basic needs, it addresses the support of the deployment of distributed applications that offer processing functionality required by its scenario or use case. This demonstrates that geoprocessing-enabled SDI can be used as the framework to run distributed geospatial applications.

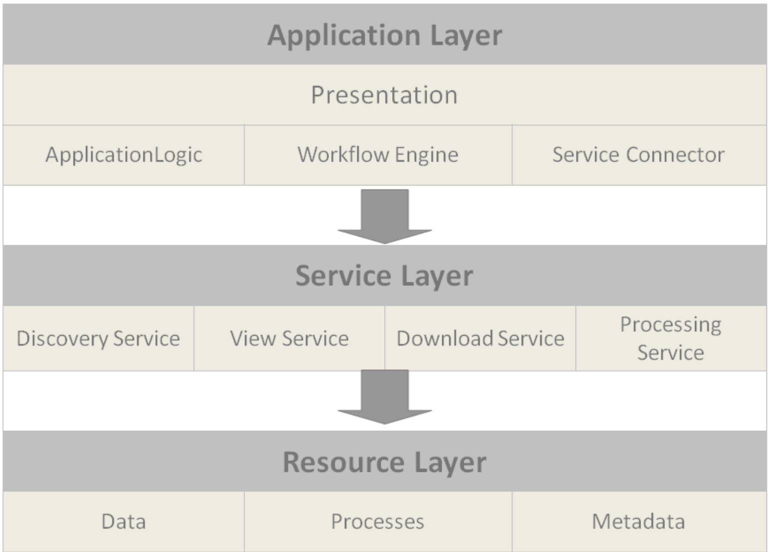

Figure 15. Conceptual architecture for distributed applications

The basic conceptual architecture is composed of the following layers:

- The **application layer** is basically composed of the presentation components containing the user interface. Distributed applications can differ in their presentation nature from a desktop application to a Web Geoportal. The remainder of the components consist of the *application logic*, the *workflow engine* and the *ServiceConnector* components. The *application logic* component implements the workflow of the application itself. This component distinguishes a centralized application from a distributed application, and it provides the orchestration of the processes that implement the application. In the case of a centralized application, the processes or steps that compose the application are within this application and they are run locally. In the case of distributed applications, both processes and the required data are distributed and placed remotely. Therefore, in distributed applications the *workflow engine* implements a service-chaining mechanism and they need to provide a S*erviceConnector* component to connect to these remote services. The S*erviceConnector* implements the interfaces to access services placed in the service layer(see below). In our case, if the INSPIRE implementation rules recommend to execute geospatial services according to standard OGC interfaces, the S*erviceConnector* has to implement the client side of this OGC interfaces.
- The core of this architecture is the **service layer**, where the services reside. These services, based on INSPIRE Service types, provide the capacity to discover, access and process data to generate and extract the desired user information. Here we emphasize the need of following INSPIRE implementation rules for the INSPIRE service types to ensure that the *ServiceConnector* will be able to connect to the services, thereby increasing the interoperability of our system. The services connect to the data layer below to provide applications to the required resources, thus acting as resource wrappers offering standard interfaces. The services at the service layer act as wrappers and mediators (Wiederhold, 1992) to integrate and offer these resources through standard interfaces, providing the system with syntactic interoperability (Díaz et al., 2009).
- The **resource layer** contains the geospatial resources stored in their original formats. Resources, in this case, are data, processes, metadata, and any other information.

This architecture addresses the needs mentioned in the previous section, including the processing capacity requirements to let users prepare information. In the first step, we consider the top-down approach where we decide to build our infrastructure based on INSPIRE principles and standards. Traditional providers (e.g. data owners, like the administration) build and deploy resources as standard services in these GII. The second step is to consider the bottom-up approach where user-driven methodologies are used. SDI users own and continuously generate data and tools. A user-driven methodology would assist users to deploy resources to the GII in order to increase their availability to build and improve distributed applications on top of SDI.

The proposed architecture does allow the existence of resources and components to run distributed applications; however, it does not contain the needed components to assist users in deploying new content to the GII. In the next section we describe an extension of this architecture to include components that assists users to deploy new content in an SDI.

3.5 Proposed extended Architecture

It is paramount that distributed resources are available in an interoperable way. Therefore, the issue is how to assist users to deploy new resources to be available in an interoperable manner. Within this context the deployment means to integrate the resources as standardized services available in the service layer according to INSPIRE principles.

The main issue is that SDI users normally do not have the knowledge to deploy resources in GII. In the remainder of the section, we present a solution to overcome this. We will describe an extension of the presented architecture to improve the availability of user resources.

First, the architecture is extended with components in which the key capacity is to assist users in wrapping resources as standard web services following the INSPIRE guidelines. Second, we propose mechanisms to improve the visibility of the deployed resources by generating metadata and publishing them in open catalogues.

The main contribution is the addition of a new module called *ServiceFramework* (Figure 16). This framework enables the users to deploy automatically resources as standard services (compliant with the INSPIRE Directive) to be available in an interoperable GII.

Furthermore, the *ServiceFramework* assists users to improve the visibility of geospatial resources by providing components to assist the users to register automatically these resources in Open Catalogues. S*erviceFramework*, is therefore, one of the main contributions of this work and it is further explained in Chapter 4.

Figure 16 shows the basic architecture extended with the *ServiceFramework* module. The Service framework is split in several layers:

- At the application layer, there is the user interface and the *ServiceConnector* components. Like we saw in the previous section, the *ServiceConnector* component allows users to connect to standard services and retrieve both data and tools.

- The *Deployer* module is placed in an intermediate layer between the application and the service layer. The *Deployer* is composed of three components: *DataWrapper*, *ProcessWrapper*, and *ServicePublisher*.
  - Process Wrapper and Data Wrapper components assist users in wrapping tools and data as standard services according to the INSPIRE implementation rules (availability).They hide the complexity of GII deployment mechanisms These two components are described in sections 4.4 and 4.5.
  - Resource Publisher aims at creating service metadata and publishing them in service catalogues (visibility). This component is described in section 4.6.
- The remaining layers of the *ServiceFramework* are analogous to the Service layer and the Resource layer in the basic architecture. The *ServiceFramework* deploys these new resources as standard services that reside in its service layer, i.e., these services now serve also these new resources. And places the new resources in its resource layer.

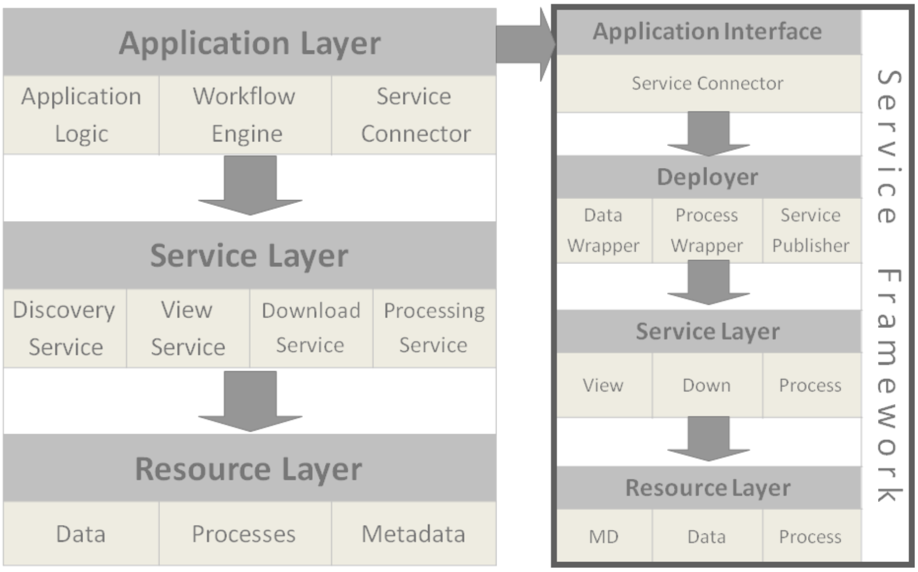

Figure 16. Extended architecture for sharing user generated resources

The *ServiceFramework* is concerned with the idea of acting as a service generator (i.e. it gets a resource and returns a standards-based service).

The design and implementation phases, described later in Chapter 4, are based on a wrapping strategy. The idea is to implement mechanisms to wrap automatically resources as standard services that guarantee interoperability. At this point, INSPIRE comes into play, providing the standards and protocols used to reach this interoperability.

3.6 Summary

We have proposed a conceptual architecture to address geospatial requirements related to running distributed applications on top of a GII. Besides general requirements like discovery, visualization and download, we have identified the following open issues and requirements to be addressed:

- Adding processing capability to GII to fulfil the requirements of modelling, extracting and preparing information.
- GII and its resources should be maintained. To successfully run these applications there is a need of plentiful resources to be publicly available. These resources should to be available in an interoperable way so we can integrate them to run distributed applications.
- End users do not have a provider role in traditional SDI because of the complex deployment mechanism and that the GII is built following a top-down methodology where only official providers can deploy new resources.

The basic architecture follows INSPIRE principles and it has been extended with our proposal: the *ServiceFramework*. In essence, this framework is concerned with the idea of acting as a service generator and publisher (i.e. it gets a resource (data or process) and returns an INSPIRE service). That service will be deployed in the GII and will then be published in a catalogue service.

The components of this architecture are described in Chapter 4 where we focus on the components and mechanisms to wrap resources (i.e. tools, data) to improve their availability and visibility. Chapter 5 will describe the use cases in which we will evaluate this architecture and its components to check the results and limitations.

4. ServiceFramework: Geospatial Resources Availability

In the previous chapter we have presented a conceptual architecture to provide geoprocessing resources to GII. In a second phase we have extended this architecture by adding new components to assist end-users to deploy resources in GII.

As shown in the literature review, most of the research done in the field of resources integration in a web environment is the adoption of web services as wrappers and mediators.

Our solutions provide mechanisms to wrap resources as standard web services, prioritizing speed using semi-automatic methods that can make great quantity of resources available by massively adding them in open public information systems.

4.1 Introduction

The current trend in information systems is the migration from monolithic to distributed environments. GIIs are experiencing the same shift (Bernard, 2003), moving to open and interoperable GIS and standardized service interfaces (Abel et al. 1998; Groot and McLaughin, 2000). As a result, the number of geospatial services available on the web is continually increasing. In spite of this, there is a lack of mechanisms to easily perform the migration to GII.

Interoperability is supposed to be ensured by a number of specification efforts, most prominently by ISO/TC 211 and OGC in the geospatial community (Bernard, 2003) and interoperable frameworks like INSPIRE. The adoption of standard services and their deployment in the middleware layer of GII makes it feasible to deploy distributed geospatial applications.

In our context we would like to distinguish between two types of services: data services and processing services. Figure 17 illustrates the idea of generating data and processing services from geospatial resources. This means that there is a need to wrap: (1) data and information to be exposed and served by standard data services and (2) tools (process, algorithms) to be exposed as standard processing services, adding processing capabilities to GII.

Processing resource availability: Wrapping software resources

Regarding the wrapping of software, Section 4.3 is devoted to describing how to deploy tools and functionality as web processing services implementing standard interfaces.

Information resource availability: Wrapping data resources

Data deployed in GII are served by standardized services. INSPIRE recommends standard interfaces to deploy them depending on their usage. The INSPIRE service types cover the user requirements when working with geospatial resources. Section 4.4 is devoted to describing how to assist the user to wrap data and information.

Geospatial resource visibility: Publishing services

To address the open issue of making geospatial resources more visible in GII, Section 4.5 is devoted to describing how standard services, serving geospatial resources, can be published automatically in open Catalogues.

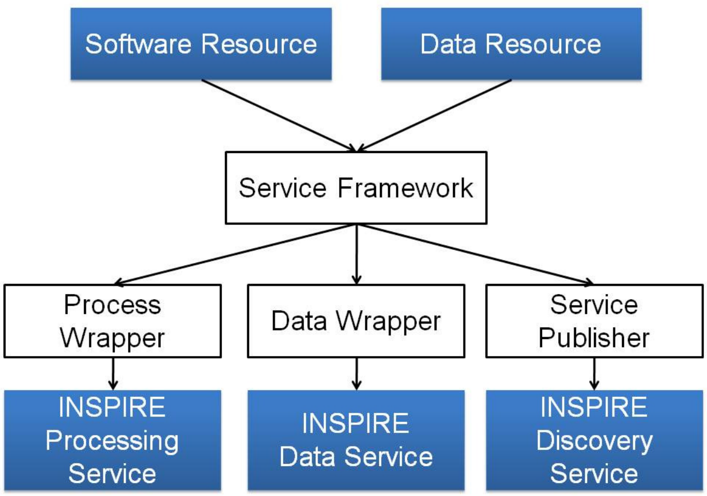

Figure 17. ServiceFramework integrating processing services generation, data services generation and service publication

The remainder of the chapter is structured as follows. Section 4.2 overviews the process development methodology that defines the different steps to approach in implementing our proposal. Section 4.3 describes the process to increase the availability of geoprocessing resources to address the requirement of providing users with reusable and interoperable processing capacity in GII. Section 4.4 proposes several components to address the requirement of assisting users to deploy data and information resources to maintain SDI and its content. Both approaches deal with assisting users by hiding the underlying technology in SDI's deployment mechanism. In Section 4.5 we propose components, deployed in our architecture, to improve visibility by generating metadata and publishing them in open catalogues.

4.2 Methodology

To develop our proposal, we will follow a software development methodology since our working framework is distributed information systems and the final goal of our contribution is to improve these systems with new software components. A software development methodology is a framework to structure, plan, and control the process of developing an information system. Figure 18 shows three software development methodologies: Prototyping, Waterfall and Spiral.

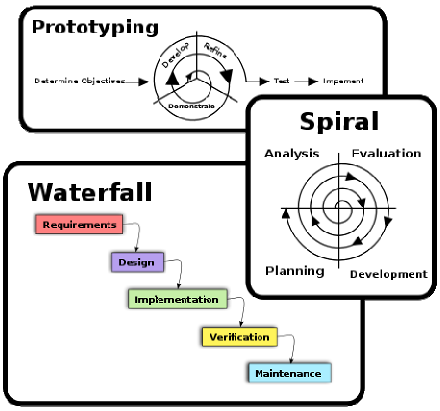

Figure 18. Three software development methodologies (Extracted from Wikipedia)

There is plenty of literature about these methodologies and it is beyond the scope of this work to discuss the advantages and disadvantages between them. The chosen methodology to develop our solution is based on the iterative and incremental development methodology (Larman and Basili, 2003) which has its steps shown in Figure 19.

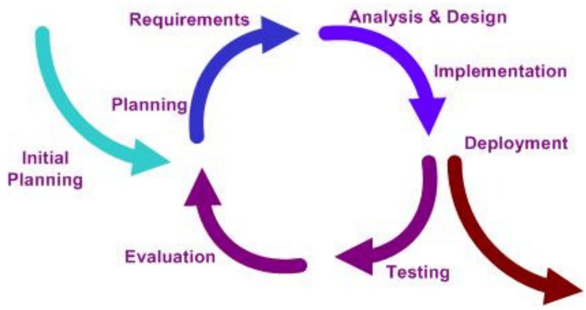

Figure 19. Iterative and incremental process (extracted from Wikipedia)

The steps that are going to structure the remaining sections are analysis, design, implementation and deployment and evaluation. Therefore, the next three sections will be structured to describe the analysis, the design of the proposed solution in terms of the components to be added to the architecture, and the implementation and deployment of the components (from Figure 17).Finally in Chapter 5, we will go deep into the evaluation of the solution applied to a use case.

4.3 Processing resource availability

This section is devoted to describing the process of increasing the availability of processing resources in GII.

We follow the iterative and incremental development methodology where the analysis, design and implementation phases of the first iteration were described. The last step of our methodology (i.e. evaluation) is performed in Chapter 5 where we describe a use case to demonstrate these applications within the context of two well-known hydrological models that have been implemented in the framework of the AWARE project. Our approach is generic enough so that it can be applied to other geosciences disciplines and areas, as we will see in a second use case provided by the European Union-funded project EuroGEOSS.

4.3.1 Analysis

The requirements that have generated the need for the addressed functionality have been identified in Chapter 3. In this analysis section the tools and functionality that may be useful in a GII for users to process and prepare their data are described. Later we determine which service interfaces would be interesting to expose these tools as processing services according to INSPIRE.

4.3.1.1 Geospatial tools

There are many software products that provide GIS functionality that are widely validated in desktop environments but underused in distributed and web environments. Normally implemented as desktop packages, they provide processing capabilities which could be migrated to geoprocessing services, thereby exposing well-tested GIS operations to distributed web services. Moreover, many of these processing capabilities have been designed and implemented by a software house for their own purposes, and so these capabilities often will not fit the needs of the concrete geospatial processing tasks performed by other user communities.

This impediment is partially being addressed by an increasing availability of diverse FOSS (Free and Open Source Software) projects which permit users to more freely choose and mix those software tools that best fulfil their requirements. FOSS projects, by the very nature of their licenses, may be modified and accommodated to suit concrete user needs. Given the wide spectrum of FOSS tools and libraries offering spatial functionality, the wrapping strategy then reflects the need of reusing existing FOSS and also closed commercial tools to wrap them as standard-conformant distributed service processes. Traditional discovery and visualization-based SDIs are evolving to a more service-based vision in which geospatial services are used to access, transform and process geospatial data.

The concept of reusability as an efficiency principle is not new. To exploit existing resources and share them between different users' needs means that:

- these resources must be publicly available instead of locally stored and they must be accessible through standard interfaces, providing the interoperability needed to be reused by other components;
- the functionality offered by services must be interesting for different user types so it can be reused in different scenarios.

The overall goal in this section is to populate GII like SDIs with tools to build distributed applications. As we are in an SOA context, the strategy consists of wrapping these tools as standard Web Services.

When designing the functionality offered by services, the following issues must be addressed:

- we have to find the right granularity to maximize reusability, and at the same time, balance the efficiency so it does not decrease due to many service calls and overload;
- we have to choose the right standard interfaces and service description to maximize their use.

4.3.1.2 Standard Interfaces

According to the INSPIRE Directive, the services deployed in an SDI should implement existing and agreed standards. OGC interfaces seem to be the chosen standards for the interfaces specification as they have been proven mature enough to become the *de facto* standards in the GIS domain.

In the related work we analyzed the standard interface candidates that will be used in the design phase. The OGC WPS is a good candidate because it describes a common interface for services offering processing operations on spatial data (vector and raster) and non-spatial data. INSPIRE does not yet recommend any specification regarding processing capacity. Therefore, we choose this standard specification to describe Web Processing Services.

4.3.2 Design

There are a few factors to consider when designing services since they become the basic computing unit upon which other components rely. First, we need to consider the granularity of the functionality provided by the service and whether it is correct for the application domain. We also have to consider other technological aspects like: (i) the situation of the software to be migrated (programming language, whether it is a library or an application, etc.), (ii) how to get a description of the available methods and the required parameters, and (iii) which service interface we want to use. These issues are described extensively in the literature and we will not cover that in this work. Our main contribution is a design methodology and a software component to assist users to design and implement processing services to improve the availability of processing resources in the SDI.

Figure 20 shows the activity diagram with the activities performed during the design and implementation phases to generate web processing services by wrapping existing functionality. The steps are as follows: First choose the functionality followed by deciding the granularity of the processes to be exposed. These two first steps belong to the design phase. The next steps belong to the implementation phase where we deal with which standard service interface to use. First, get the signature or description of the functionality regarding the required parameters. Wrap it as services according to the chosen interface and deploy it. And finally the generation of metadata to publish in catalogue services to increase the visibility of the newly generated services.

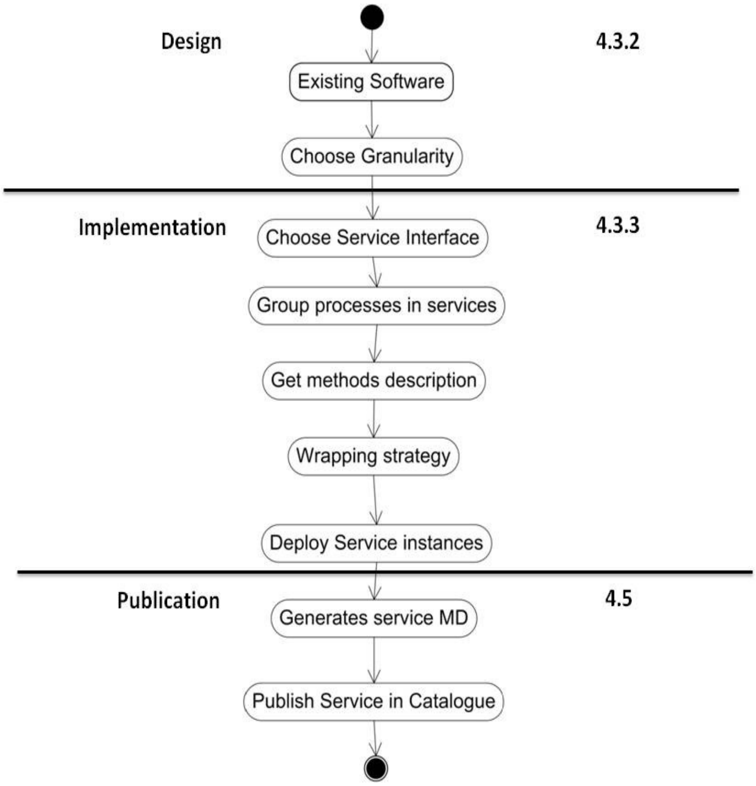

Figure 20. Activity diagram: Design and implementation steps for integrating existing or new software as WPS in GII

4.3.2.1 Software (processing functionality)

As INSPIRE Directive mandates, SDIs should contain in their service layer service types that allow stakeholders to discover, visualize, access, and download resources. Due to user requirements, SDIs have increasingly evolved from its only-access-and-visualize capabilities to more advanced processing-enabled SDIs through the deployment of geospatial processing services. Because we focus on processing functionality, we next describe how to design processing services to be optimally deployed in an SDI.

Processing functionality requirements depend on the application scenario and the use cases. Therefore we do not analyze concrete functionality but we describe the next steps of design and implementation assuming an abstract functionality. To validate the design and implementation process, we will analyze concrete functionality requirements and apply our solution in the use cases to be presented in Chapter 5.

4.3.2.2 Granularity

Service design principles in SOA seek to minimize strong coupling to guarantee that services are self-contained, modular, extendable and reusable (Papazoglou and Yang, 2002). Creating services for specific application requirements implies the necessity to find the right level of granularity which has been pointed out as a main issue in the SDI research agenda (Bernard et al., 2005). Service granularity refers to the size of a service in terms of the amount of functionality carried out (Haesen et al., 2008).

Coarse-grained services encapsulate larger groupings of capability within a single interface, reducing the number of service requests from the client necessary to accomplish a task. However, as a downside, they might return excessive quantities of data, or it might be difficult to modify them to meet new requirements. Fine-grained services are those which perform specific tasks that are no longer decomposable into smaller pieces (subject to the application requirements). Small services normally require less complicated input and output data, meaning that they can be composed more easily.

Having a small, stable set of coarse-grained services is often considered a best practice in designing services in SOA. However, we prefer to consider the full spectrum of granularity levels, from coarse-grained to fine-grained services, in order to show how geospatial services at different granularity levels might have a positive impact on service reusability and performance. Finding the adequate granularity is a matter of balancing multiple criteria (flexibility, modularity, reusability, and performance) to meet the ongoing needs of a specific application (Feuerlicht, 2006; Haesen et al., 2008). Coarse-grained services usually offer better performance, however, their flexibility decreases to adapt to new requirements. Creating fine-grained services that can be easily reused in other workflows is our goal but we must craft the right balance of fine-grained and coarse-grained services to meet the ongoing needs in our context. The strategy to design services and their granularity is in most cases tied to the particular use case or scenario. We describe several strategies and we apply them in the use cases in chapter 5.

Top-down Strategy

This strategy is a good approach when, for instance, a group of scientists split a scientific model into smaller pieces of functionality to identify the most relevant processes. This recursive methodology continues until we encounter the desired level of granularity for the given processes. The criterion to stop the top-down decomposition is designed to check if a given process is specific, yet functional enough to not be split again. This decision stems from a consensus between service designers and scientists.

Each of these processes is exposed within a processing service. These processes that implement a small and basic functionality have a potentially high degree of reusability. This allows them to be integrated as part of the other workflows in the same application and/or in different domains. For example, if we expose a process that calculates the area of a polygon, it could be used a number of times by the same application and by other applications.

After these basic functions are exposed as processes, we may find a combination of processes that are executed together and in the same order, like a chain of tasks (Voisard and Schweppe, 1998). A recursive practice in application development in GIS and SOA is to combine elementary functions as more complex tools to meet specific requirements, and therefore, create coarser-grained services. This reduces network traffic and overload when calling different services.

Bottom-up Strategy

A bottom-up strategy is a good approach when, independent from a specific scenario, we have a set of tools developed as a software library and we want to expose it as distributed services. The strategy then consists of wrapping these tools with the implicit granularity. Afterwards, if desired, we can develop coarser-grained services by combining and chaining these processes as we previously did before.

In the first use case within the AWARE project, our proposed strategy has followed a top-down methodology. We will explain this in detail in Chapter 5.

4.3.3 Implementation

In this section we describe the activities that belong to the implementation phase shown in Figure 20 as well as the components involved in these activities.

4.3.3.1 Service Interface

We have based our architecture on the INSPIRE architecture and principles (see Chapter 3), and therefore, we try to generate services according to the INSPIRE Service types to maximize the interoperability with related systems. As mentioned in the analysis section, INSPIRE recommends the use of existing standards and the IIR recommends to use of OGC standards.

According to the INSPIRE technical guidelines, *Transformation Services* only transform coordinates, whereas *Invoke Spatial Data Services* is able to invoke and chain services. For this reason, we have decided to extend INSPIRE types and create another service type: *Processing Service Type*. A similar approach is taken in the ORCHESTRA project (ORCHESTRA, 2008). In this section we design and generate processing services to be placed under this *Processing Service Type*.

Currently, IIR define standards for the View, Download and Discovery service types. Since we extend an INSPIRE service type to create our *Processing Service Type*, there is no clear recommendation for us to choose a concrete standard. Because the new OGC WPS specification fits our needs and considering the IIR and the OGC influence, this specification is chosen to generate processing services (see OGC WPS in related work).

OGC Web Processing Service 52North Implementation

After the analysis of several OGC WPS implementations, we chose the implementation of 52North for the use cases described in Chapter 5. The main reasons are that it is written in Java, it is open source, and it turned to have a short learning curve during the analysis phase. As a result we could implement a WPS in a relatively short time. 52North WPS supports WPS 0.4.0 and 1.0.0 versions. Version 1.0.0 offers the functionality to expose geoprocessing functionality as OGC- and W3C-compliant services.

This Java framework for developing an OGC WPS has an extensible architecture for processes and data encodings based on the notions of repositories. It also provides dynamic access to the embedded functionality of the WPS already registered in the framework (Foerster, 2006). This extensible framework allows us to add new processes to its core. It implements the WPS interface where we add our own processes that were developed in Java. Process (an operation with inputs and outputs) is the basic operational unit of the OGC WPS.

Most of the functionality needed in our use cases is already implemented as desktop applications or libraries. The goal is to transform these libraries into processes available in a 52North WPS instance. Figure 21 shows how we can add new processes to a 52North WPS instance. To add a new process, we have to implement it in a Java class that is extended from the abstract class called *AbstractAlgorithm*.

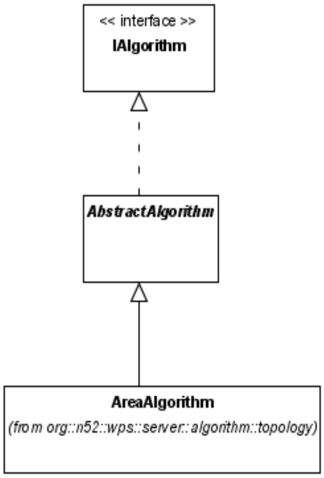

Figure 21. Adding a new process using 52North OGC WPS implementation

Besides adding the functionality encoded as a new Java class under this hierarchy, there is a need to provide a process description to the inputs and outputs required to run the process. This description is called the *service signature* and it contains details about the process, such as what it does and which parameters are necessary to execute it. Thus this description is necessary to remotely execute the process. According to the OGC WPS standard, each process of the WPS should provide a *describeProcess* document compliant with the WPS schemas (Schut, 2007) in response to a *describeProcess* request. This means that the code migrated to the WPS should provide a way to request this information so this process can be generated in a programmatic fashion. Otherwise, this document has to be generated manually by the programmer.

When we add new processes to a WPS instance, we can see these processes served by a WPS through querying the service capabilities document (see Figure 22 below).

```
<?xml version="1.0" encoding="UTF-8"?>
<Capabilities version="0.4.0" xsi:schemaLocation="http://www.opengeospatial.net/wps
D:\_work\specs\wps\WPSDRA~1\wps\0.4.0\wpsGetCapabilities.xsd"
xmlns="http://www.opengeospatial.net/wps" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <ows:ServiceIdentification>
        <ows:Title>UJI topology WPS 0.4.0</ows:Title>
                <ows:Abstract>This service provides access to processes through the common
WPS interface</ows:Abstract>
                <ows:Keywords>
                        <ows:Keyword>Processes</ows:Keyword>
                        <ows:Keyword>UJI, SP</ows:Keyword>
```

```
                <ows:Keyword>Topology </ows:Keyword>
            </ows:Keywords>
            <ows:ServiceType>WPS</ows:ServiceType>
            <ows:ServiceTypeVersion>0.4.0</ows:ServiceTypeVersion>
        </ows:ServiceIdentification>
        <ows:ServiceProvider>
            <ows:ProviderName>Universitat Jaume I</ows:ProviderName>
            <ows:ServiceContact>
            <ows:IndividualName>Laura Diaz</ows:IndividualName>
            <ows:PositionName>phd</ows:PositionName>
            <ows:ContactInfo>
            <ows:Address>
            <ows:City>Castellon</ows:City>
            <ows:Country>Spain</ows:Country>
            <ows:ElectronicMailAddress>laura.diaz@uji.es</ows:ElectronicMailAddress>
            </ows:Address>
            </ows:ContactInfo>
            </ows:ServiceContact>
        </ows:ServiceProvider>
        <ows:OperationsMetadata>
            <ows:Operation name="GetCapabilities">
                <ows:DCP>
                    <ows:HTTP>
                        <ows:Get
xlink:href="http://localhost:8080/wps/WebProcessingService"/>
                    </ows:HTTP>
                </ows:DCP>
            </ows:Operation>
            <ows:Operation name="DescribeProcess">
                <ows:DCP>
                    <ows:HTTP>
                        <ows:Get
xlink:href="http://localhost:8080/wps/WebProcessingService"/>

                    </ows:HTTP>
                </ows:DCP>
            </ows:Operation>
            <ows:Operation name="Execute">
                <ows:DCP>
                    <ows:HTTP>
                        <ows:Post
xlink:href="http://localhost:8080/wps/WebProcessingService"/>
                    </ows:HTTP>
                </ows:DCP>
            </ows:Operation>
        </ows:OperationsMetadata>
<ProcessOfferings>
<Process>
<ows:Identifier>org.n52.wps.server.algorithm.AreaAlgorithm</ows:Identifier>
</Process>
</ProcessOfferings>
</Capabilities>
```


Figure 22. Example OGC WPS *GetCapabilities*

Apart from getting the available processes by means of the capabilities, we can request a description of a process to run it. To configure a WPS in 52North, we provide and register the algorithms classes, and we provide an XML file that describes the new algorithm in terms of its inputs and outputs. That XML file is the *describeProcess* document that we will retrieve after we send a WPS *describeProcess* request. Figure 23 shows an example of this document:

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:ProcessDescriptions
xmlns:wps="http://www.opengeospatial.net/wps"><wps:ProcessDescription processVersion="1"
storeSupported="true" statusSupported="false" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:xlink="http://www.w3.org/1999/xlink">
        <ows:Identifier>org.n52.wps.server.algorithm.AreaAlgorithm</ows:Identifier>
        <ows:Title>Calculates FeatureCollection area.</ows:Title>
        <ows:Metadata xlink:title="spatial"/>
        <ows:Metadata xlink:title="geometry"/>
        <ows:Metadata xlink:title="GML"/>
        <wps:DataInputs>
            <wps:Input>
                <ows:Identifier>data</ows:Identifier>
                <ows:Title>Polygon</ows:Title>
                <wps:ComplexData defaultFormat="text/XML"
defaultSchema="http://schemas.opengis.net/gml/2.1.2/feature.xsd">
                <wps:SupportedComplexData>

    <wps:Schema>http://schemas.opengis.net/gml/2.1.2/feature.xsd</wps:Schema>
                </wps:SupportedComplexData>
                </wps:ComplexData>
                <wps:MinimumOccurs>1</wps:MinimumOccurs>
            </wps:Input>
        </wps:DataInputs>
        <wps:ProcessOutputs>
            <wps:Output>
            <ows:Identifier>result</ows:Identifier>
            <ows:Title>area</ows:Title>
            <ows:Abstract>area output</ows:Abstract>
             <wps:LiteralOutput>
            <ows:DataType ows:reference="xs:double"/>
            </wps:LiteralOutput>
            </wps:Output>
        </wps:ProcessOutputs>
    </wps:ProcessDescription></wps:ProcessDescriptions>
```

Figure 23. Example OGC WPS describeProcess

The *describeProcess* gives us the information to build an execute request to send to the WPS. An example of an execute request is shown in Figure 24.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Execute service="WPS" version="0.4.0" store="false" status="false"
xmlns="http://www.opengeospatial.net/wps"
xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xsi:schemaLocation="http://www.opengeospatial.net/wps
..\wpsExecute.xsd" xmlns:om="http://www.opengis.net/om"
xmlns:gml="http://www.opengis.net/gml">
<ows:Identifier>org.n52.wps.server.algorithm.AreaAlgorithm</ows:Identifier>
  <DataInputs>
    <Input>
      <ows:Identifier>data</ows:Identifier>
      <ows:Title>input</ows:Title>
      <ComplexValueReference ows:reference="http://localhost/aware/finalTest/basin.gml"
schema="http://schemas.opengis.net/gml/2.1.2/feature.xsd"/>
    </Input>
   </DataInputs>
</Execute>
```

Figure 24. Example OGC WPS Execute request

An example of an OGC WPS execute response containing the result of the process is shown in Figure 25.

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:ExecuteResponse xmlns:wps="http://www.opengeospatial.net/wps">
<ows:Identifier xmlns:ows="http://www.opengeospatial.net/ows">
org.n52.wps.server.algorithm.AreaAlgorithm </ows:Identifier>
<wps:Status>
<wps:ProcessAccepted>This process was executed at: 12-mar-2008</wps:ProcessAccepted>
</wps:Status>
<wps:OutputDefinitions>
<wps:Output>
<ows:Identifier xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">result</ows:Identifier>
<ows:Title xmlns:ows="http://www.opengeospatial.net/ows"> Calculates FeatureCollection
area.</ows:Title>
</wps:Output>
</wps:OutputDefinitions>
<wps:ProcessOutputs>
<wps:Output>
<ows:Identifier xmlns:ows="http://www.opengeospatial.net/ows">result</ows:Identifier>
<wps:LiteralValue dataType="xs:double">25085.0</wps:LiteralValue>
</wps:Output>
</wps:ProcessOutputs>
</wps:ExecuteResponse>
```

Figure 25. Example of an OGC WPS execute response

4.3.3.2 Components

Figure 20 is an activity diagram where the actors or components involved in each activity are not shown. This section describes these components that have been introduced in Chapter 3.

Figure 26 shows the architecture described in Chapter 3. The components, highlighted with a darker colour, play a special role and represent a contribution of this work for increasing the availability of processing resources. After the methodology to design the processing services, we describe which components are related to each task.

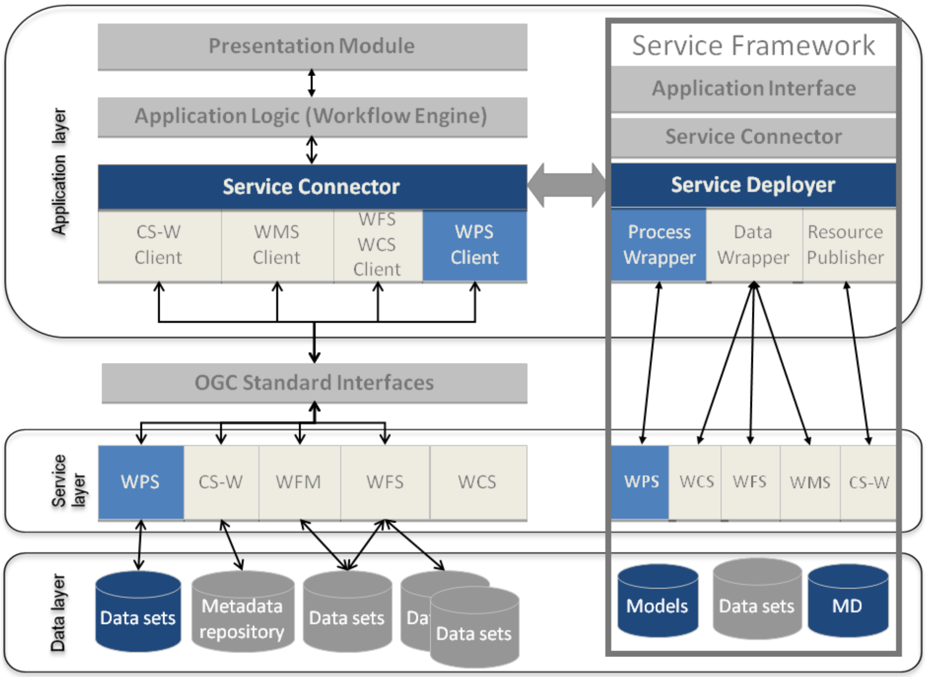

Figure 26. Extended architecture with details on the components involved in increasing the availability of processing resources

ServiceConnector

The ServiceConnector component in the application layer contains components that implement the client interfaces of the standard services deployed in the service layer. The Web Processing Service Application Programming Interface (WPSAPI) implements the OGC WPS client interface to be able to access and run distributed processing services.

WPSAPI

The proliferation of service-based applications to realize the full potential of SOA demands mechanisms to allow client applications to invoke services anywhere and retrieve their results in an easy way. The WPSAPI component pursues this goal because it is responsible for instantiating, connecting, and executing remote, distributed geoprocessing services. It connects to geoprocessing services interfaced by the OGC Web Processing Service.

The WPSAPI component provides a set of interfaces to communicate and interchange information with remote (WPS-based) services. This WPSAPI is a self–developed Java library which implements the client interface of the OGC WPS specification. It offers a programmatic way to send and receive *getCapabilities* request using Java objects. It implements objects to send and receive Process description requests to get the description of processes in order to see the parameters and the way a process within a WPS has to be invoked. Finally, it permits sending an execute request to invoke one of the available processes and it is able to retrieve and parse the response in order to extract the process results. We can see the class diagram of the WPS client component in Figure 27. The object model is scalable because it would support new versions of the WPS specifications with minor changes. Currently versions 0.4 and 1.0.0 are supported, and it is easily scalable to support future versions.

Figure 27. WPSAPI class diagram

Service-Layer: WPS

In the service layer of our architecture, we deploy the OGC WPS that we have designed and implemented according to the decisions mentioned in the previous section. We will evaluate and demonstrate the implemented OGC WPS in the AWARE project use case in Chapter 5. The component related to this task is the *ProcessWrapper* within the *ServiceFramework*.

ServiceFramework

The *ServiceFramework*'s architecture is a multilayer architecture as we introduced in Chapter 3. It is an extension of the proposed architecture to add components regarding the improvement of the availability of geospatial resources. Next we describe the components related to adding geoprocessing resources.

Service Deployer

Within the *ServiceFramework,* we find the *ServiceDeployer* module. The *ServiceDeployer* assists the user to generate and deploy services from resources. This module contains three main components (Data Wrapper, Process Wrapper, and Service Publisher). In this section we describe the *Process Wrapper* component whereas the *Data Wrapper* component will be described in Section 4.4 where we deal with increasing the availability of data resources and Section 4.5 will describe the Service Publisher.

Process Wrapper component

The *Process Wrapper* component deals with the wrapping of software processes as standard processing services, in particular as OGC WPS. In its current status, it is not a reusable component but a proof of concept of the methodology to add processing capabilities to SDI. The main components are shown in Figure 28.

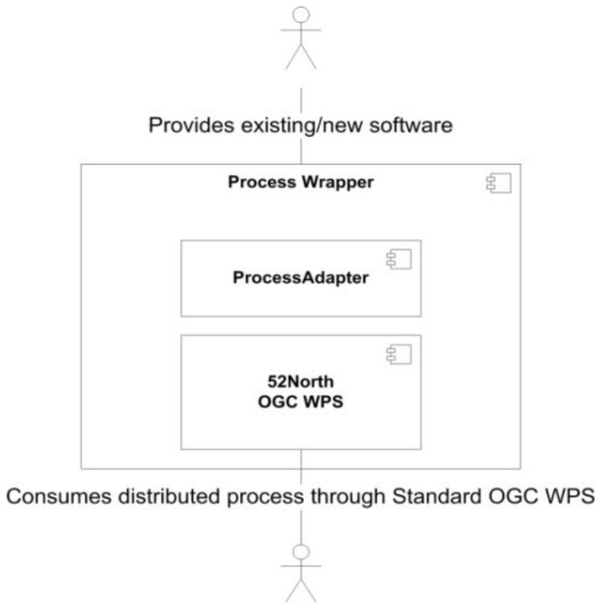

Figure 28. Process Wrapper Components diagram

As described, the implementation of the OGC WPS specification provided by the 52N WPS framework enables the deployment of Java algorithms into WPS services. The *Process Wrapper* is implemented in the two components shown in Figure 28. First, the *Process Adapter* component adapts the required functionality to generate 52N WPS-enabled processes.

As we can see in Figure 29, the *Process Wrapper* is integrated in the 52N WPS framework. The basic functioning consists of aggregating a new process to a WPS. The approach is further discussed. Using the design pattern *Adapter* (Gamma et al., 1995), we create an Adapter implementation for each process that has been selected to be wrapped and deployed in a WPS. We aggregate it to the 52N framework as a class that implements the 52N *AbstractAlgorithm*. Afterwards, this adapter delegates the WPS request to the corresponding software which also makes the corresponding data conversion. As mandated by the OGC WPS specification, each process must have a *describeProcess* document for invocation.

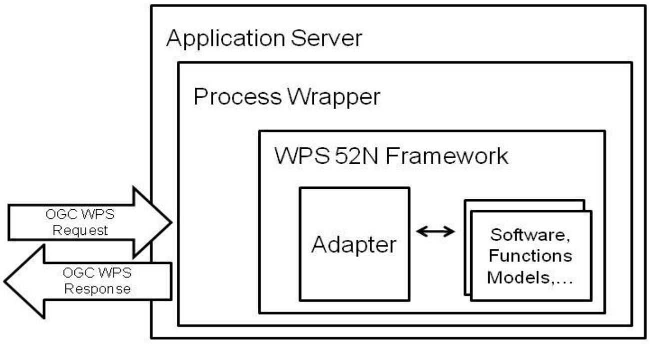

Figure 29. Process Wrapper delegator architecture

4.3.3.3 Wrapping Strategy

The WPS implementation is based on wrapping. Most geospatial packages provide processing capabilities which can be migrated to geoprocessing services, thereby exposing well-tested GIS operations to web access as distributed services. Given the wide spectrum of FOSS tools and libraries offering spatial functionality, the wrapping strategy reflects the need of reusing (mostly) existing FOSS but also closed commercial tools in order to wrap them as standard-conformant distributed service processes.

In the use cases, several FOSS tools have been reused merely to the extent of creating the service interface, leaving the original implementation unchanged. In other cases, modifications of the tool code were necessary to adapt them to our needs. In the cases where we needed functionality that was not available, we implemented it from scratch utilizing available FOSS tools such as GeoTools, gvSIG, SEXTANTE, and JFreeChart.

As an example, the GeoTools library supports the implementation of the processes concerned with topological tasks involving the calculation of geometric area and intersection. The JFreeChart library, which provides an API to create charts and line diagrams, has been integrated in the Chart geoprocessing service to deliver the diagram functionality required in our application. In the same way, the SEXTANTE library provides a set of raster and vector analysis operations.

In other cases, scientists have scientific routines already implemented in software modules (e.g. interpolation routines) using specific IDL and Fortran libraries which were not suitable for an easy migration to distributed web environments. These legacy routines were wrapped using dynamic libraries and Java bridges to expose the embedded functionality as processing services.

The wrapping approach is the approach followed in our use cases to wrap existing functionality as OGC WPS. This is developed within the *Process Wrapper* component. Currently, this is only a partially-reusable component because it has been developed as a proof of conc,ept within a concrete scenario, of the methodology to add processing capabilities to the SDI, providing potentially reusable standard processing services.

To illustrate the implementation of one of these services based on 52N WPS, we describe *Sextante* WPS, one of the WPS created in the AWARE use case (see section 5.1). *Sextante* WPS actually wraps SEXTANTE operations. Figure 30 shows the relationship among processing service types, specifications (WPS, WSDL/SOAP, etc.) and concrete implementations (52N WPS, pyWPS, etc.). The figure further shows the steps carried out to implement the *Sextante* WPS from its abstract definition down to the concrete realization exposing processes like *vectorize, classification, etc. as OGC WPS.*

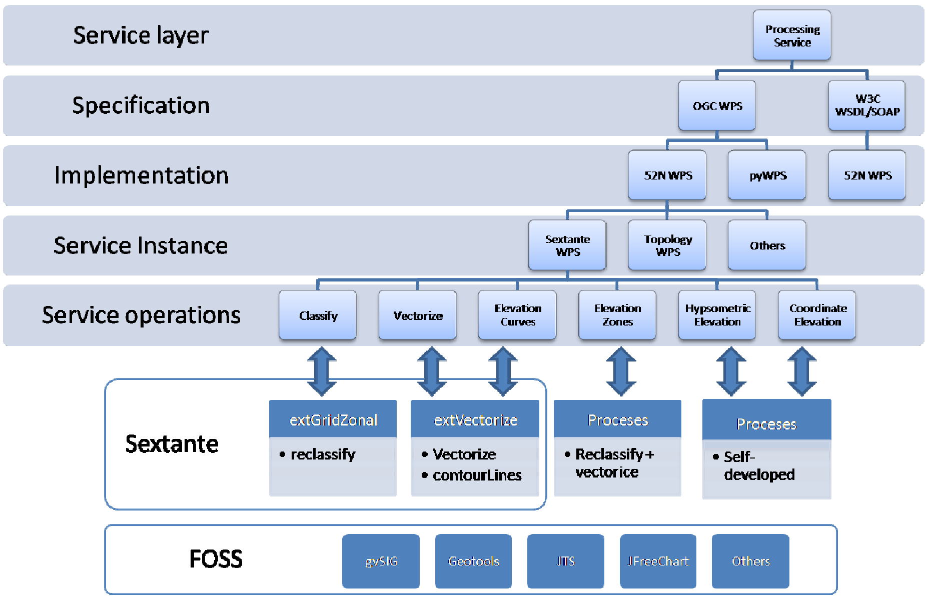

Figure 30. Example of decisions from abstract definition to implementation using Sextante WPS (extracted from AWARE use case)

Each SEXTANTE (Olaya, 2007) extension represents a single analysis process and it is based on a so-called Algorithm/Extension architecture which formally separates the processing (the algorithm) itself from creating the user interface or handling output data, among others. This architecture is particularly suitable for exposing SEXTANTE algorithms through a WPS interface. SEXTANTE algorithms contain two main methods (*defineCharacteristics()* and *processAl-*

gorithm()) which are equivalent to WPS operations (i.e. *DescribeProcess* and *Execute*).

To wrap SEXTANTE into the 52N WPS framework in a way that allows the easy addition of functionality to SEXTANTE when needed, we have followed the Adapter pattern (Gamma et al., 1995) as illustrated on the left side of Figure 31. The right side of Figure 31 shows the class diagram for the adapter pattern implementation. The abstract class *AbstractAlgorithm* must be extended when adding a new process to a 52North Service. In this way all the algorithms created in 52N that expose SEXTANTE functionality will implement the Adapter pattern, thereby having an instance of the *SextanteAlgorithm* to be invoked later. When the process (52N algorithm instance) is executed, it will call the correct method in the *SextanteAlgorithm* interface called *ProcessAlgorithm* (Díaz et al., 2008).

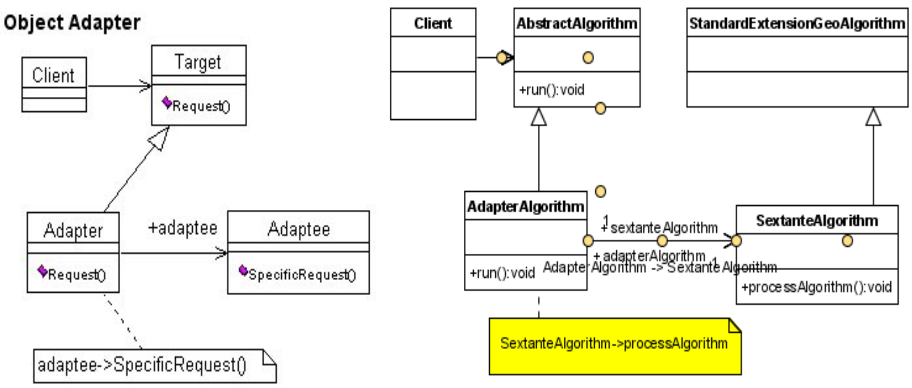

Figure 31. Adapter pattern. The left side depicts generic patterns and the right side depicts the pattern implementation applied to semi-automatic migration from SEXTANTE algorithms to WPS.

4.4 Information resources availability

Besides policies and rights management, exposing data resources through standard services is a matter of technological knowledge. Normally, resource providers must have this know-how knowledge to deploy and maintain services to serve their data and information. The insufficient knowledge provokes a lack of rapid ways to share crucial and up-to-date information with other stakeholders and places many impediments to rapid availability and visibility of geospatial resources

SDI stakeholders process data continuously to generate new information that is normally used to generate reports. In contrast to the current situation, this user-generated information should be persisted so that the outcomes could be easily shared and reused later. Due to this dynamism and the constantly updating requirements of geospatial resources (particularly in disaster management scenarios), we would like to let users, who are in touch with the management situation, add and update data and information to the GII so they become readily available for sharing. Therefore, we would like to assist with the migration of these unconnected data and newly generated information to a collection of standardized services that are accessible, for instance, via a web-based entry point. This is similar to the philosophy of Web 2.0 where users are provided with tools and mechanisms to facilitate information sharing. Issues like security and quality are raised and they will be discussed later in this chapter.

GII users are ready to shift to the consumer-provider role, thereby actively participating in the development and maintenance of the GII. There is a need for tools and mechanisms that allow GII users to integrate data resources in a way where they do not have to be aware of the complex underlying technology. The participation of users may induce a massive addition of data resources which is a need for this new generation of applications to succeed.

In this section we address a number of issues related to the generation and publication of data services providing data and-user generated information. Our contribution is two-fold: (i) using the presented architecture and distributed application to generate new information by means of standard services, and (ii) providing mechanisms to assist users to wrap data resources (data and information). These mechanisms will generate and publish automatically standard web services in SDIs serving these resources.

4.4.1 Analysis

The requirements have been identified in Chapter 3. In this section we describe the geospatial data and information that may be useful to be available in the GII as well as service interfaces to which exposing these data resources as distributed data services would be interesting.

4.4.1.1 Geospatial data and information

Huge amounts of data, available from multiple sources and stored in multiple formats, need to be organized so that they can be accessed and exploited efficiently. Leaded by international initiatives, governmental regulations and public demands, the public and private organizations are willing to serve geospatial information resources – data and processes — on an interoperable basis to save money and human resources, and add value to the environmental information according to spatial data infrastructure principles of discovery and reuse. However, the complexity of implementing these services impedes their wide availability, limiting access to up-to-date information during critical situations such as in disaster response.

As shown in Figure 32, in addition to the availability of raw data, users are generating new information every time they process these data by using geospatial applications. Nowadays, intermediate or final process and modelling results are used mainly to generate reports; unfortunately, they are not persisted in an interoperable way by being deployed back to the GII.

Figure 32. Information generation process

As described in previous chapters, the tendency to share and reuse geospatial resources is by means of standard services in the SDI. Therefore, to turn SDI in a bi-directional platform, we need to assist users with automatic manners to deploy resources by wrapping them as standard and interoperable services on the SDI.

4.4.1.2 Standard Interfaces

The INSPIRE technical architecture (Figure 4) shows that the different service types defined by the Directive are placed in the service layer resulting in the so-called *Service Network*. Each type defines common capabilities offered by a group of services. Specific service types like discovery services offer end users a common mechanism to search discoverable geospatial data. In this section, we will focus on the Data Services. These services let users access, view and download data. Regarding this functionality, the INSPIRE Service Types we consider are the *View* and *Download* Service types.

OGC Service implementation specifications

The IIR recommends the use of wide spread standards like OGC. We have described the following OGC specifications regarding data services that are relevant: OGC WMS, OGC WFS and OGC WCS (see Chapter 2 for more information).

4.4.2 Design

In the previous section, we discussed methodologies and design criteria to implement mechanisms that increase the availability of interoperable tools. In this section, we apply the same methodologies to design a solution to increase the availability of data resources in the GII.

Figure 33 shows the activity diagram with the steps that we go through during the design and implementation process to generate web data services by wrapping data resources. First, users choose the data or information they want to share. Second, we need to decide by what means we want to expose the data. In this case, we have limited it to two functions of view and/or download. These first two steps belong to the design phase and will be described in detail in the following sections. The next steps belong to the implementation phase where we decide which standard service interface to use. In the last steps we deal with deployment details and the generation of metadata to publish services in Catalogue Services in order to increase the visibility of these generated services.

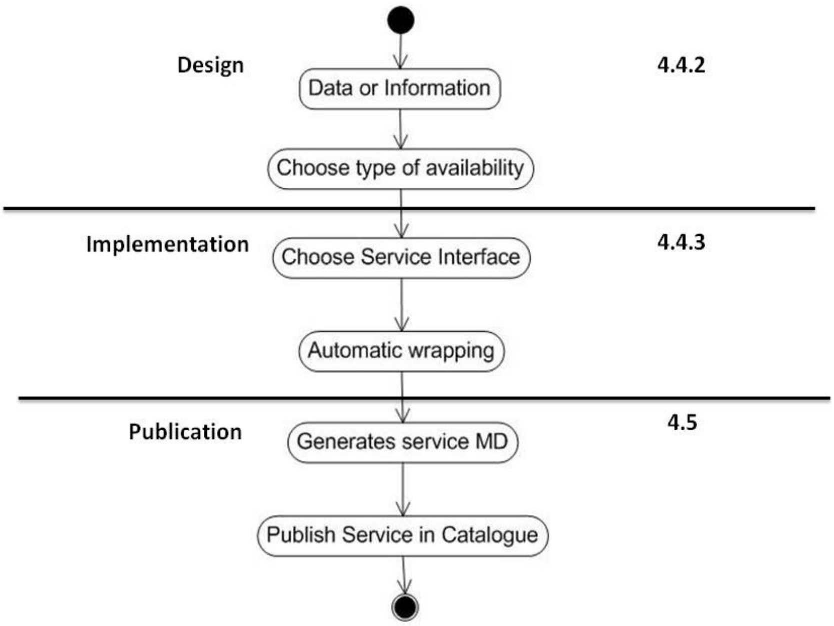

Figure 33. Activity diagram: Activities in the design and implementation steps for integrating data resources as standard data services in GII

4.4.2.1 Data and information types

We have briefly mentioned the differences between *sharing data* and *sharing information*, defining information as the data that have been processed to extract some information about an issue. With this methodology we let SDI stakeholders share data and information. We will provide mechanisms for the user to pick up a raw dataset such as primary data in raster and vector format. We also provide mechanisms so the users can process data to extract information, which they can then share. Since our purpose is to integrate data resources according to the INSPIRE technical guidelines, the proposed solution assists users to deploy data as INSPIRE Service types classification (i.e. visualization and download purposes). The service types according to INSPIRE classification are View Service Type and Download Service Type.

The automatic deployment of data resources as these INSPIRE Service types is currently implemented in the proof of concept. In the future, limiting user requirements can be specified as well, such as limiting the data to be expose only as View Service type.

View Services

INSPIRE asks Member States to "establish and operate view services making it possible, as a minimum, to display, navigate, zoom in and out, pan or overlay viewable spatial data sets and to display legend information and any relevant content of metadata"(Service Network DT, 2009).

Download Services

INSPIRE asks Member States to "establish and operate a network of download services, enabling copies of spatial data sets, or parts of such sets, to be downloaded and, where practicable, accessed directly"(Service Network DT, 2009).

4.4.3 Implementation

In this section we describe the activities that belong to the implementation phase shown in Figure 33 and the components involved in these activities.

4.4.3.1 Service types Interfaces

According to INSPIRE IR, the standard specifications to be implemented by the services instances of the view and download category are as follows:

View Services

The base specification of an INSPIRE View Service is ISO 19128:2005(E) WMS international standard (Service Network DT, 2008).

Download Services

It is recommended by INSPIRE to implement the direct access services using the WFS as specified in ISO/DIS 19142 and with the query facility of Filter encoding (FE) as specified in ISO/DIS 19143. These versions of WFS and FE were jointly developed by OGC and ISO/TC 211 and represent the latest versions of the specifications Implementing Rule on interoperability of data sets and services or technical guidance. The pre-defined data set or pre-defined part of data set shall be encoded in GML as described by ISO 19136 (Service Network DT, 2009).

If the functionality of the web coverage service is required for a given theme (like raster data), then the candidate normative reference is OGC 07-067r5 OGC Web Coverage Service (OGC WCS) Implementation Standard 1.1.2 (Service Network DT, 2009).

To summarize, the standard specifications to be used in each case to integrate resources in the GII are:

Vector data	View (WMS) Download (WFS)
Raster data	View (WMS) Download (WCS)

Table 1. OGC specifications to deploy data resources in SDI

OGC Standard Implementation

There are many implementations of the interfaces OGC WMS, OGC WFS and OGC WCS. In this thesis, Geoserver is the implementation of choice. Geoserver offers the implementation of the three interfaces and it provides a Java API that lets us configure and deploy services programmatically. Geoserver API lets us wrap data resources by using the GeoServer concept for wrapping resources called data stores. This API is an important component used in the Data Wrapper component.

4.4.3.2 Components

In Chapter 3 we have presented an architecture in which to deploy components to run distributed applications and to increase the availability of resources. In the last section we proposed design criteria to increase the availability of data and information and migrate them to web data services. In this section we describe the components of the architecture that allow this design and provide the necessary mechanisms.

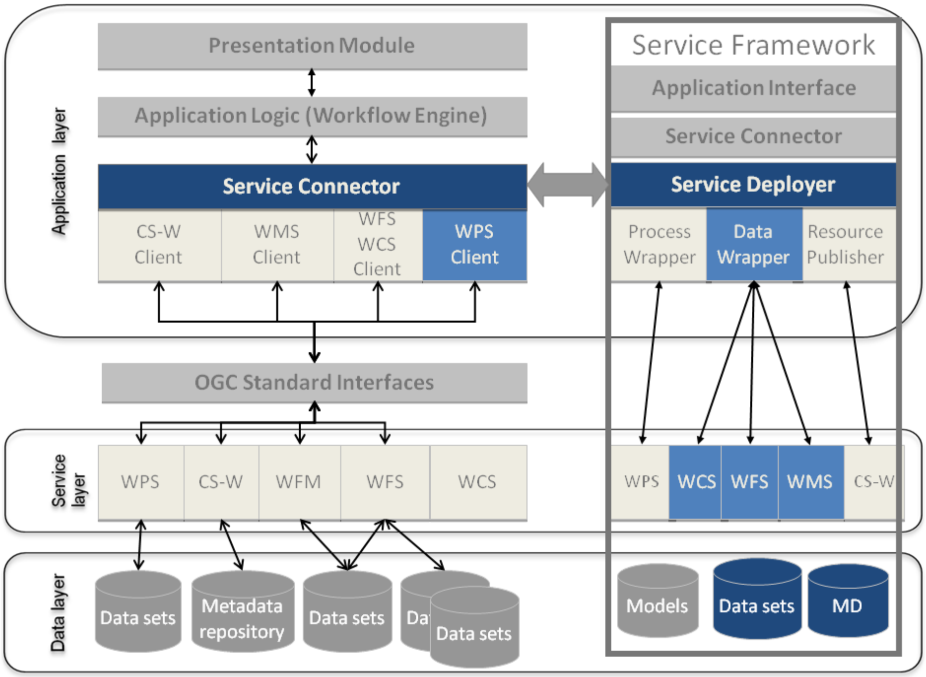

Figure 34. Architecture and components with details for data wrapping

Figure 34 shows the architecture where we have highlighted with a darker colour the components that play an important role and represent a contribution of this work for the wrapping of data and information to be deployed as standard data services on top of an SDI.

In general, the *ServiceConnector* in the application layer contains the components to connect and access the deployed data services (according to INSPIRE Service Types and IR). But in this section, we want to describe the write-mode (rather than read-mode) of these data services and generate and add content to data services using the *ServiceFramework*. On one hand, users are able to publish data as services since the *ServiceFramework* provides an interface to assist users in this task. On the other hand, users can also publish newly generated information. The WPS Client component within the *ServiceConnector* implements the OGC WPS client interface to access and run processing services in the service layer. Processing services generate new information that can be sent to the *Service Deployer* in the *ServiceFramework* and then deployed as data services by the *Data Wrapper* component according to the INSPIRE Service Types *view* and *download*.

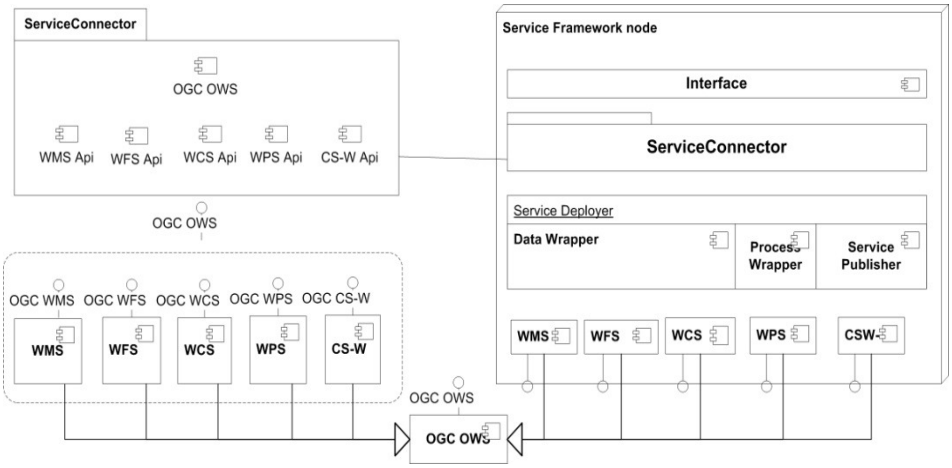

Figure 35. Conceptual components diagram

Figure 35 shows the conceptual components and the interfaces implemented by each of them. Next we describe these components and their implementation details.

ServiceFramework

The *ServiceFramework* architecture is a multilayer architecture. On the top layer we find the *Interface* component that provides the functionality to capture data and user-generated information to be wrapped. The data being wrapped can be raw data or processed data. ServiceFramework let users to process data by connecting to WPS for running a process and generate new information. The Wrapper Module will assist users in converting these valuable outputs to standard data services according to the INSPIRE IR. This aspect is very useful for disaster management because this facilitates to deploy and integrate the new generated information maintaining up-to-date the information in the GII. Thus, to let users select and add newly generated information, we have the *ServiceConnector* that lets users process data and retrieve information by means of the WPS API.

The user interface of the *ServiceFramework* has been developed using JavaFX technology[xxix]. From a development perspective, the *ServiceFramework* follows the Model View Controller (MVC) design pattern that integrates a simple viewer using OpenLayers[xxx] technology. It is further described in Chapter 5 within the EuroGEOSS use case.

WPS API

The WPS API component provides a set of interfaces to communicate and interchange information with WPS-based services. This component lets users generate new information by accessing and processing data. This self–

developed Java library implements the client interface of the OGC WPS specification and lets us retrieve processed information. When a user generates new information by means of this API, he or she will be able to wrap this information as a service. The reason is that the *ServiceConnector* can be integrated within the *ServiceFramework* and connect to the data wrapper, thereby creating the mechanism to increase the availability of interoperable data and information.

ServiceDeployer module

Within the *ServiceFramework* we find the *ServiceDeployer*. The ServiceDeployer assists the user in automatic or semi-automatic ways to generate and deploy the services from resources. This module contains three main components. The main goal in this section is to increase the number of data and information to run in the distributed applications, and this task is performed with the *DataWrapper* component.

DataWrapper component

As we have mentioned, the *DataWrapper* component deals with the wrapping of data and information to create standard services concretely as View and Download INSPIRE Service types, thus implementing OGC WMS, OGC WFS and OGC WCS according to INSPIRE IR and the data nature (raster or vector) as we will see in the implementation section. The main components are the components shown in Figure 36.

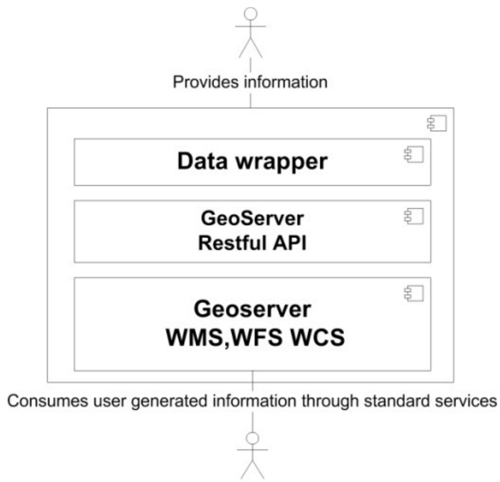

Figure 36. Data wrapper detailed components

The *DataWrapper* component deals with the availability issue pointed out earlier. This module provides a mechanism to wrap geospatial data as standard services using OGC WMS. OGC WFS and OGC WCS both distribute data in their original formats and let users download them. On a technical level, this component is implemented by the recent RESTful API from Geoserver.

The *DataWrapper* component retrieves the data or the information retrieved by the interface, or the *ServiceConnector*. Finally, the data wrapper layers provide the functionality to turn data resources into OGC service interfaces by using the Geoserver RESTful API component. *DataWrapper* generates a new data source in a Geoserver instance (that resides in the service layer of the ServiceFramework) This Geoserver instance implements the interfaces of OGC WMS, OGC WFS and OGC WCS. The ServiceFramework creates a new data source deploying the data resources to be served by this Geoserver instance. After the new services have been generated or updated, they are then deployed or updated in the service layer within the *ServiceFramework*.

4.4.3.3 Wrapping Strategy

Our strategy to increase data and information availability in SDI environments is based on wrapping, specifically in automatic wrapping. We distinguish between wrapping data and information to show how in the latter some of the components developed in section 4.1 to generate information play an important role and are linked with the *DataWrapper* component.

Data wrapping

The steps to wrap data resources and the accompanying components are shown in a sequence diagram (Figure 37). First of all, the user selects the data that he or she wants to publish and this is done in the *ServiceFramework* interface or in the interface of the application where the *ServiceFramework* is integrated. The *ServiceFramework* component in charge of starting the data wrapping is the *DataWrapper* component which is situated within the *ServiceDeployer.* This component uses the Geoserver RESTful API to add a new data source in the Geoserver instance, which is associated with the *ServiceFramework*. After adding this new data source, the *DataWrapper* component configures a new layer and coverage as well as a set of features according to the data nature (raster or vector data). By default, this new dataset is added to a View Service type or OGC WMS and to a Download Service type (i.e. an OGC WFS or OGC WCS for a vector or raster data, respectively).

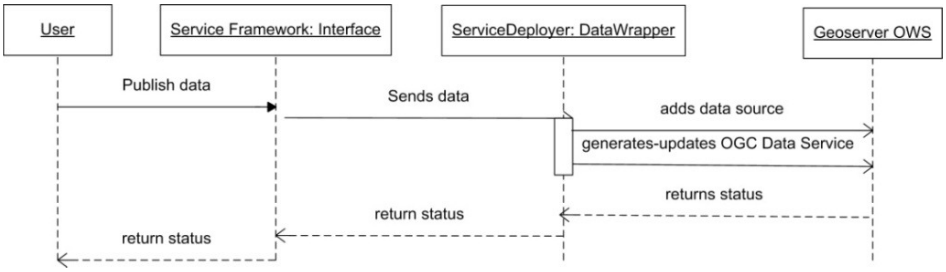

Figure 37. Sequence diagram for data wrapping

The next screenshots show the prototype developed for the proof of concept. Figure 38 shows the interface to assist the user in selecting the data for publishing (pertaining to the first sequence in Figure 37).

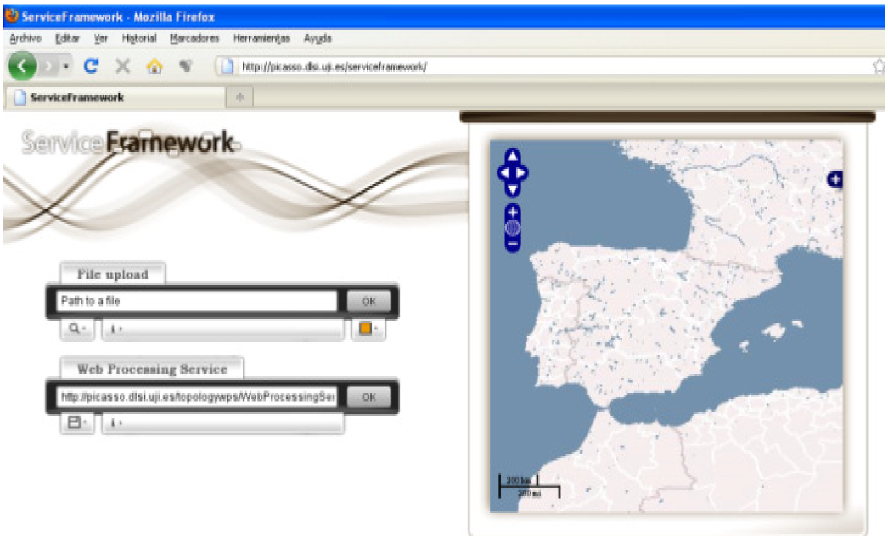

Figure 38. ServiceFramework interface regarding the data wrapping

Figure 39 shows the consequence of the second and third sequence where the *ServiceFramework* sends the data to the *DataWrapper* which adds a new data source in the Geoserver. Figure 39 shows a new layer being updated in the Geoserver instance within the *ServiceFramework*.

Figure 39. Data published in the Geoserver instance in *ServiceFramework*

Figure 40 and Figure 41 show how the data published in the standard services can be retrieved by other interoperable means like OGC WMS clients (e.g. gvSIG[xxxi] or uDIG[xxxii]). The screenshoots were created with the uDIG software.

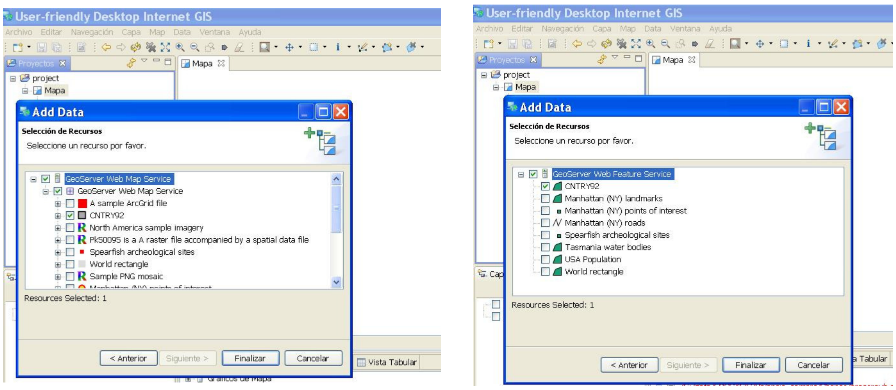

Figure 40. Retrieving data from uDIG OGC WMS and WFS client

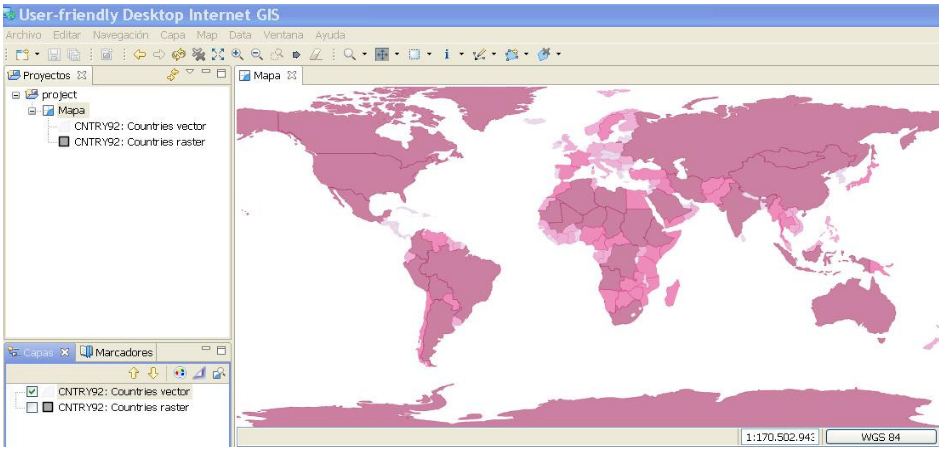

Figure 41. uDIG screenshot shows how published data in OGC services can be retrieved by interoperable OGC clients

Newly generated information wrapping

As we have mentioned in the earlier sections, we defined *information* as data that have been processed in order to extract information required to solve a problem and to find some answers in a certain scenario. Therefore, the strategy followed to wrap information is similar to wrapping data, but it includes a few prior steps that help the user process data and generate this information. Figure 42 shows the sequence diagram where we can see the components involved in each step of the user-generated information wrapping.

First of all, a user can select or add a new WPS in the *ServiceFramework* interface or in the interface where the *ServiceFramework* has been integrated. The communication with the WPS is done by the WPS API component within the *ServiceConnector*. A previous step not shown in the sequence diagram is the *getCapabilities* request (to the chosen WPS) to inspect the available processes. After the user has selected a process, the WPS API sends a *DescribeProcess* request to the WPS to retrieve the required inputs and outpus to execute the process. At this point, the user must select the data according to the process inputs. The input data resources can be retrieved from local storage or accessed through remote, standard download services. In the latter case, the ServiceConnector is used to request these remote data. Once the data inputs are provided, the user can execute the process assisted by the *ServiceFramework* interface. Once again, the WPS API in the ServiceConnector sends an execute request to the WPS and retrieves the generated information that will be shown to the user in the map viewer.

One scenario of a user generating new information to be deployed in the GII by using the *ServiceFramework* can be as follows: A user is executing a Buffer process which needs a polygon in GML format and an integer. This GML data can be provided by the user (i.e. from a local file) or the user can request a WFS serving the required polygon. In the latter case, the WFS client in the *ServiceConnector* will retrieve the desired features from the selected WFS.

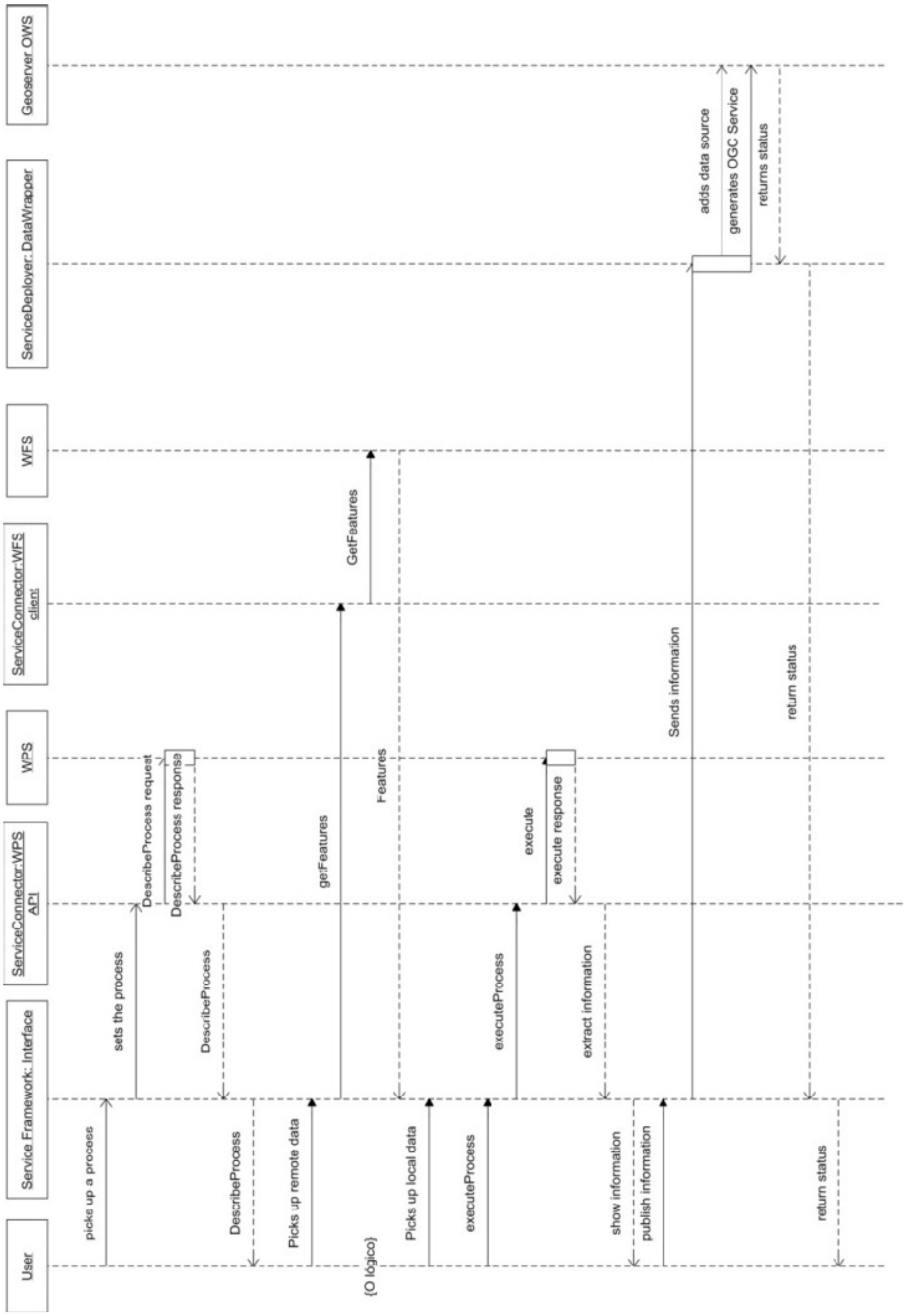

Figure 42. Sequence diagram for user-generated information wrapping.

Once the data inputs are provided, the user can execute the process that is assisted by the *ServiceFramework* interface. The WPS API in the *ServiceConnector* sends an execute request to the WPS and retrieves the generated information that will be shown to the user. This newly generated information can be published immediately and be available through the interoperable services to be reused in other scenarios. To publish the newly generated information, the process is performed by the *DataWrapper* component as described in Figure 37.

We can conclude that both data and intermediate processing results (wrapped as WMS and WFS) can be visualized in the uDig and gvSIG desktop clients or in any WMS-WFS-compliant client.

4.5 Service Publisher

Improving visibility of geospatial resources means to enhance the capability for the resources to be found. Because of the current status of the SDI and the mechanisms deployed for discovery, metadata (necessary to describe the resources) and Catalogue Services are the key elements for the discovery and resource fusion possibilities (Nogueras et al., 2005; Díaz et al., 2007). In this context, the INSPIRE Directive mandates the creation and maintenance of metadata and related Discovery Services (Craglia et al., 2007). According to the INSPIRE IR, Discovery Service types need to provide the functionality to find and access resources according to the traditional model of *publish-find-bind.* These services must implement the OGC Catalogue Service for the Web (CS-W) standard interface in several of their specific profiles.

In Chapter 3 (see Figure 14) we saw the life cycle steps of a geospatial resource. Publication and Discovery steps are related to this section. Resource Publication is about publishing metadata in open and standard Catalogues while Resource Discovery is about being able to connect and query these Catalogues to access the metadata containing the description of the resources regarding what they are and how we can obtain them.

After the proposed solution to increase resource availability to assist users to integrate and update content in a GII, we now address the Resource Publication. The main goal is to assist users in publishing what they have integrated in the GII, and thereby increase the visibility of these resources.

There is an extensive literature about metadata creation tools and methods. It is broadly accepted that metadata creation is a complex process that is normally separated from the data creation process and it involves an extra cost for data providers (Manso et al., 2004; Gould 2006a; Gould 2006b; Gould et al., 2006; Rajabifard et al., 2009). These are some of the reasons of the scarcity of metadata that provokes a poor knowledge of the existence of data (Craglia et al., 1999; Official Journal of the European Union, 2003; Nogueras et al, 2005).

We would like to introduce the distinction between the metadata of data and the metadata of the services. Along this work, we propose to integrate spatial data by wrapping and deploying it as standard services. Although we remark the need of data metadata in the SDI and their current status quo, our contribution in this section is limited to the publication of the generated services.

4.5.1 Analysis

Regarding the publication, first we should consider the most extensive way to find and bind resources in an SDI. Nowadays, Catalogue Services are the mechanism of reference to find and bind geospatial resources in the GII. Catalogues Services support both data and services. Therefore SDI stakeholders query these Catalogues in order to find data resources or services deployed in the SDI. We have proposed the integration of resources in the SDI by means of wrapping them as standard services. We would like to close the resource life cycle by designing and implementing a component within the *ServiceFramework* which assists users in publishing the generated services. In this section we focus on the service metadata and how we publish it in Catalogues Services.

4.5.1.1 Service Metadata

It is not a goal of this work to discuss the many different ways and languages of describing general web services. We will discuss briefly how currently geospatial services are described in the context of an SDI.

One of the most broadly used metadata standard for resources is ISO 19115:2003. In 2005, ISO/TC211 defined an extension of it to create a standard metadata for service which is now known as ISO 19119:2005(7).

4.5.1.2 Standard Interfaces

OGC Catalogue Service for the Web (CS-W)

This service interface specification is the interface commonly used in SDI to implement Catalogue Services. This interface specifies methods to query the catalogue and retrieve the results including the metadata of the resources.

INSPIRE Directive has also adopted in its Implementing Rules this interface as the interface to be implemented by the Discovery Type Services (INSPIRE DT, 2009b).

OGC Web Services (OWS)

According to the OGC OWS (Whiteside, 2005) interface, all the OGC Services must provide a method in their interface called *GetCapabilities* that retrieves an XML-based document called Capabilities which contains service metadata describing the service identification, provider, operations, content, etc.

4.5.2 Design

We describe a solution to generate and publish the metadata of service resources in Catalogue Services in existing SDIs in a semi-automatic way. The *Resource Publisher* component, illustrated in the architecture section, provides this capability.

Figure 43 shows the activity diagram with the general steps that we went through during the design and implementation process to publish a service in a service catalogue. First of all, users can select one of the generated services to request the OGC Capabilities of this service and the generation of metadata. Second, the user can select a catalogue service in which to publish this service. A connection to this catalogue is open, the metadata are sent, and it concludes with the publication of the service.

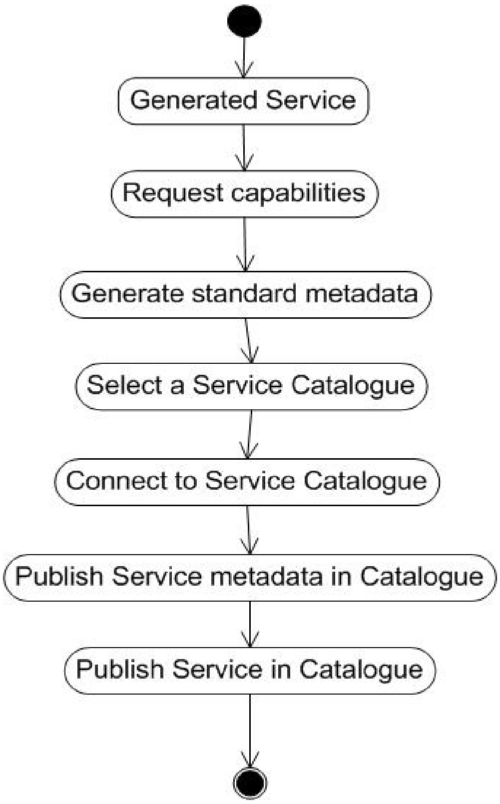

Figure 43. Overview on the general steps in the design and implementation for integrating data resources as standard data services in SDI

4.5.2.1 Metadata

Generate resource metadata

It is out of the scope of this work to discuss about the automatic generation of metadata for geospatial data (Bulterman, 2004; Manso et al., 2004). Remains as an open issue of this work to collect and exploit metadata in a more transparent, scalable and less tedious part of the overall workflow (Díaz et al., 2007). There are software packages that are capable of extracting a minimum set of metadata elements of the most common formats for geospatial data and creating metadata in a standard format (ISO19139, Dublin Core, etc.). For example, the metadata manager portion of the open source GIS gvSIG (Díaz et al., 2008) could potentially be integrated in the Service Publisher component and it will be considered in future work.

We will focus on the metadata of the generated services, like the metadata for the WPS. Besides the *GetCapabilities* document, the WPS interface has a *describeProcess* request which returns also an XML-based document with metadata that describes the inputs and outputs for the process to be invoked remotely. This document is the OGC-like WSDL document, thus it plays a central role in the publish-find-bind paradigm (i.e. it can be used for the process to be registered, found and executed). It is out of the scope of this work to collect this metadata, but it is also considered as future work.

Generate service metadata

According to the OGC OWS (OGC Web Services) (Whiteside, 2005) interface, all the OGC Services must provide some service metadata in the Capabilities document. The open source OGC implementations (Geoserver and 52North) that we have used to wrap the resources (data and models) implement the OGC specification so that they are able to create the OGC Capabilities document.

Ongoing collaboration with the metadata group in the Spanish SDI is being done to transform automatically these service metadata to other standards like INPIRE. The automatic generation is based on the matching between capabilities and other OGC documents and INSPIRE implementing rules for metadata specifications.

4.5.3 Implementation

In this section we describe the activities that belong to the implementation phase and the components involved in these activities.

4.5.3.1 Service types Interfaces

According to the Technical Guidance for INSPIRE Discovery Services (NS DT, 2009), the standard specification to be implemented by service instances is the discovery category OGC-CSW.

INSPIRE Discovery Service Type

The base functionality of an INSPIRE Discovery Service is derived from CSW ISO AP. The INSPIRE requirements as defined by INS DIS and CSW ISO AP distinguish between the two types of catalogue services: a *read-only* catalogue service that has to provide operations labelled with 'CSW' and a *transactional* catalogue service that has to provide operations labelled with 'CSWT'. This distinction is derived from the OGC catalogue base specification (OGC CSW).

In our context, the *ServicePublisher*'s goal is to publish service metadata in one of the Discovery Service instances. Therefore, the Discovery Service would need to implement a transactional interface in order to publish metadata. At the current state in the proof of concept, we did not use the CS-W transactional interface but we chose to implement our own interface to do the transactional operations. In this way, the CS-W catalogue client (implemented within the *ServicePublisher*) does not currently implement the CS-W transactional interface as it is left for our future work.

Discovery Service Type Implementation: GeoNetwork

The implementation of the OGC CS-W specification is provided by GeoNetwork[xxxiii]. GeoNetwork open source is a standards-based Free and Open Source catalog application.

4.5.3.2 Components

In the previous sections we have proposed a methodology that allows users to increase the availability of resources by migrating them to web services. In this section we want to describe with more detail the components of the architecture that allow users to publish these services in Open catalogues so they can be further discovered and bound

We have seen in previous sections the components diagram of the *ServiceFramework* which is the main contribution of our work and which contains the components that implement the proposed solution for the issues highlighted in this section. The *ServiceFramework* lets users integrate new resources in the GII and, as we will explain here, lets users publish these services in Service Catalogues

ServiceFramework

ServiceFramework is composed of different layers as shown in various figures throughout this work. The top layer contains the interface component that provides the functionality to interact with the user, and in this section we will discuss how it lets the user select the service to publish in the target catalogue.

ServiceConnector

The *ServiceConnector* component provides a set of interfaces that allows communication and the interchange of information with OWS-based services. This component retrieves the Capabilities document from the services to be published. This component contains the WPS API component which is a self–developed Java library that implements the client interface of the OGC WPS specification. Currently, *GetCapabilities* is the only method implemented across the other OGC client interfaces in the ServiceConnector. *GetCapabilities* is required to get the service metadata and it is crucial in this section.

Service Deployer module

Within the *ServiceFramework* we find the *ServiceDeployer*. The *ServiceDeployer*, among other things, assists the user in the automatic publishing of services. This module contains three main components. In this section we are going to describe the *ServicePublisher* component.

Service Publisher component

As we have mentioned, the *ServicePublisher* component deals with the publishing of service metadata in Catalogues Services. And as concretely as Discovery INSPIRE Service type, it implements the OGC CS-W according to the INSPIRE IR. The main components are shown in Figure 44. The interface component interacts with the user who selects an OGC service to be published. The *ServiceConnector* requests the Capabilities document of the selected service. The Metadata Transformer should be able to transform the Capabilities document to other metadata standards such as the INSPIRE metadata specifications. Catalogue Client, which conceptually should be included as part of the *ServiceConnector,* implements the Catalogue service client interface to be connected to the target catalogue. The resources are finally published to potentially be discovered later by all the users in the information system.

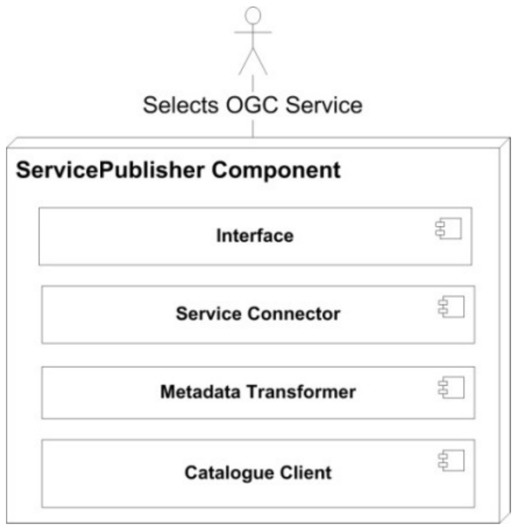

Figure 44. Service Publisher Component Diagram

4.5.3.3 Publication Strategy

The last step in our methodology deals with this automatic publication of the generated services. Thus our proposed mechanism implemented by the *ServicePublisher* component provides the functionality to connect to a Catalogue service, validating the user and publishing the service automatically. Figure 45 shows the steps to publish the generated OWS. First of all, the user selects the OGC Service. Second, the *ServiceFramework* connects to the OWS by using the *ServiceConnector* to obtain the service metadata and descriptions within the *OGC Capabilities Document*. Afterwards the *ServicePublisher*-Metadata Transformer transforms this service metadata, if needed, to other supported metadata standards. Finally, the Catalogue Client connects to the Catalogue Service to perform authentication and validation (for rights to publish), and finally publishes the services in a Catalogue Service. All of the steps are transparent to the user.

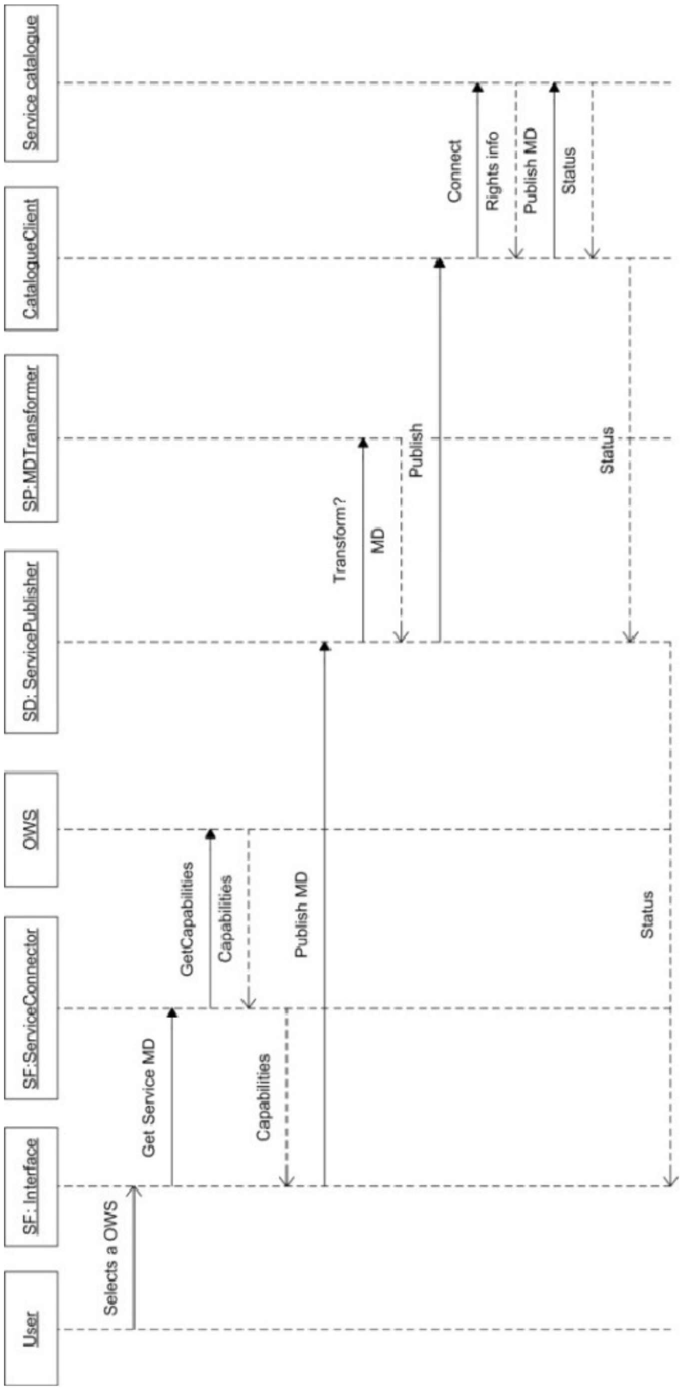

Figure 45. Sequence diagram of publishing OGC Services in Service Catalogues with the ServiceFramework

For the first prototype we have chosen GeoNetwork as the implementation of the Catalogue Service. The OGC CS-W standard specification is the standard adopted by the INSPIRE IR to be implemented by the Discovery service types. The *ServicePublisher* publishes the metadata using the Geonetwork protocol. For future work, we are extending the prototype to use other OGC CS-W (Nebert and Whiteside, 2004) transactional implementations.

Figure 46 shows the *ServicePublisher* interface to assist the user in the automatic publication of services in the Catalogues Services. The user simply types the Catalogue Service URL and the URL of the OGC Service to be published.

Figure 46. Service Publisher interface (proof of concept)

4.6 Summary

In chapter 3 we described general and specific requirement from different perspectives. They have been the starting point to address the analysis, design and implementation of the proposed architecture and its components, in particular, the ServiceFramework. To describe how the proposed solution achieves the mentioned requirements we conclude with a brief summary of the addressed issues:

Processing resources availability

The overall goal in this section has been to improve the availability of processing resources in the GII to provide users with reusable tools to build applications and process data in a distributed fashion. This alleviates the need to maintain multiple desktop software packages for the purpose of a few occasional operations. The unstructured methodology of scientists, decision-makers and other SDI stakeholders using different scientific desktop tools, data and algorithms is migrated to a collection of standardized services accessible in an interoperable way (for instance, via a web-based entry point).

The added value is that the scientific processes wrapped as standard web services can be reused in other scenarios. These applications can be dynamically configured and created by chaining geospatial services described using standard interfaces adopted by INSPIRE Directive. The final goal is to encourage the model web as a new paradigm for scientists working in a distributed and remote environment in order to reuse and share geospatial resources. At the end we will describe the challenges to be faced to apply this in a real case scenario.

Information resources availability

The second goal has been to improve the availability of data and information resources to make GII more useful as a work environment. Our approach has been to add a new role to end users where they could not only consume but also provide their knowledge and information to this GII. Providing mechanisms to hide the underlying technology and allowing users to massively add data and information as standard services increases the amount of interoperable resources in GII.

Service visibility

Regarding the visibility of the new generated services, we have introduced a limited approach. A minimum set of metadata elements, describing the generated services, has been automatically published in an open Catalogue. Therefore although the metadata are very simple the services are published in the standard catalogue without much effort from the user.

5. Evaluation: Experiments on the Use cases

5.1 Experiments on the AWARE project: Generation and deployment of processing services to run scientific models.

This use case is being carried out in the framework of the European-funded FP6 project called AWARE. In the project we have developed a web based application for running runoff forecast models, permitting expert users (hydrologists and other scientists) to calibrate and run models and permitting non-expert users to access the results.

This use case purpose is twofold: on one hand, it demonstrates the architecture based on INSPIRE principles presented in Chapter 3, and on the other hand it demonstrates the proposed methodology in Chapter 4 to design and implement processing services.

5.1.1 Analysis

The AWARE Application is able to run two hydrological models. Together with a hydrologist team, we have analyzed the functional requirements of these models in order to identify the relevant tasks in their workflow to ultimately expose them as Web Processing Services.

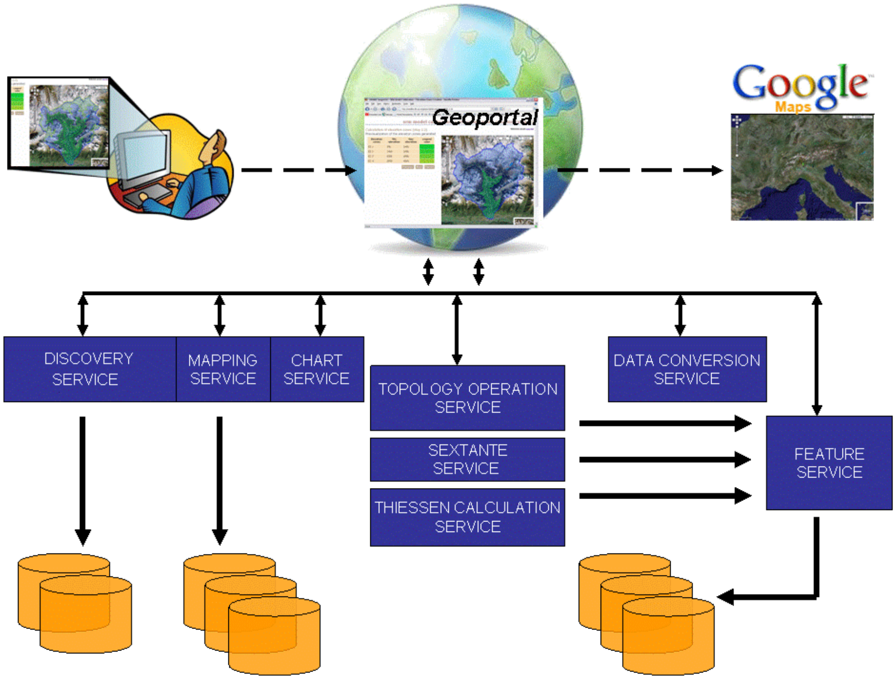

Figure 47. General use case

Figure 47 shows a schematic diagram of the generic use case proposed in this work. We have added processing capabilities to the SDI, deploying a set of processing services on the service layer. Users can access them though the entry point of their applications, and in this case, a scientist connects to the Geoportal. Although in the project the Service Layer contains other INSPIRE Service types such as discovery and download, we focus on the processing services that we have designed and deployed to evaluate their ability to provide distributed and reusable processing capacity.

The hydrological models are implemented as predefined workflows in the Geoportal. The notion of workflow treated here is a set of tasks that can be executed without user supervision, as a predefined chain of tasks. The hydrological model logic component within the Geoportal permits the execution of the scientific workflows that shape the hydrological model. The component dictates how the services have to be invoked within each scientific workflow and it acts as an orchestration module (i.e. it will run the hydrological models as successive calls to the service processes available in the AWARE service network). As defined by ISO standard 19119 (2005), the kind of chaining performed in this case is called *opaque chaining*, where the user invokes the Geoportal which acts as an aggregated service that carries out the chain.

In Figure 48, we can see the workflow of calculating a basin area. The predefined chain will first call the discover service to find the data of the basin. The second step will ask for a View Service type to render the data for visualization in the Geoportal. The third step is to call a data access service type to retrieve the vector data of the basin. It describes the input in Geographic Mark-up Language (GML) format. If the basin data is in a different format (e.g. provided by the user in shapefile format), the Geoportal will call the Data Conversion Service which provides a Shp2GML process. The fifth step calculates the basin area using the *AreaCalculation* process that is available in the Topology Service.

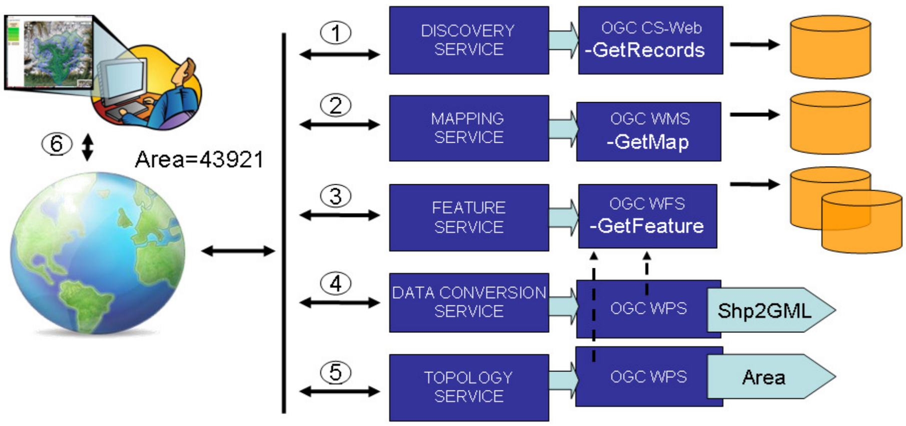

Figure 48. Area calculation use case

5.1.2 Architecture

The AWARE Application is a distributed, web-based application for running hydrological models in the realm of the SDI and the INSPIRE vision.

Figure 49 illustrates the AWARE architecture. This architecture is based on the proposed conceptual architecture in Chapter 3. In this use case we particularly adapt some layers like the Geoportal layer. In the AWARE application the Geoportal enables users to interact with the hydrological models and visually explore the model results. The hydrological model logic includes service chaining control and then allows the instantiation and invocation of services by using the ServiceClient module which corresponds to the ServiceConnector described in Chapter 3 and 4. As described previously, the Service layer combines a set of service instances grouped in service types which are based in the INSPIRE service types. Finally at the bottom, the Data layer contains spatial datasets and metadata.

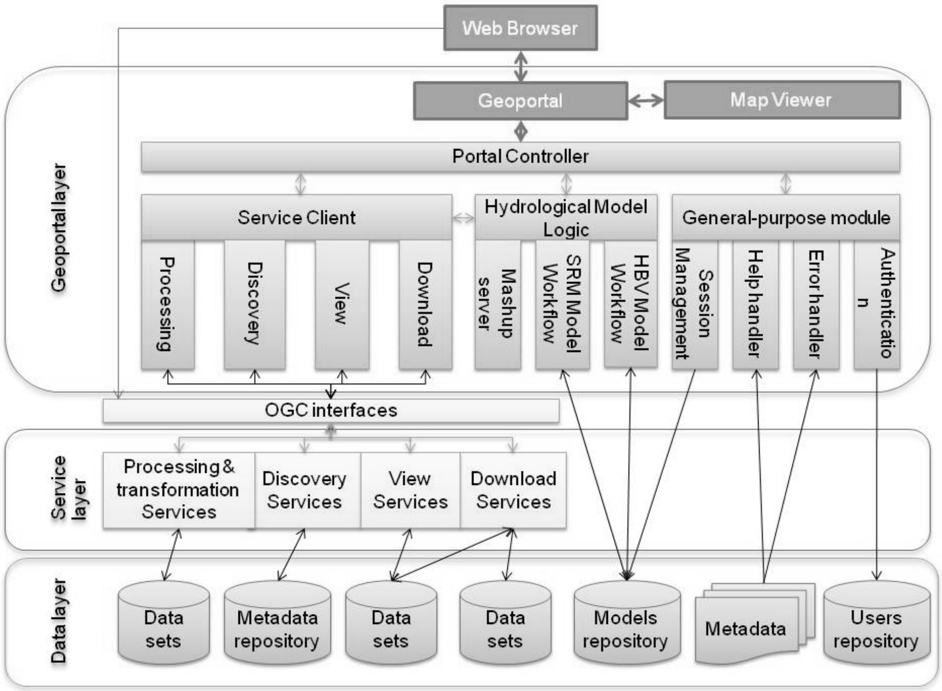

Figure 49. AWARE architecture based on the proposed architecture described in Chapter 3.

5.1.2.1 User interface and data visualization

It is out of the scope of this dissertation to go through the details of the AWARE Geoportal layer. We briefly describe (as shown in Figure 49) that users access the application via a web browser which connects to the Geoportal module. The Geoportal layer configures and runs environmental models for a particular watershed of study. The AWARE application (compared with other environmental modelling tools and systems discussed in the related work section) does not require any special software packages and desktop GIS systems on the client side. The only required software in the client machines is a web browser with an internet connection. This is an example of emerging client solutions which are tailored to certain workflows, and are thus flexible and inexpensive in terms of software licensing (Moreno-Sanchez et al., 2007).

5.1.2.2 Service integration and interoperability

The ServiceClient module corresponds to our ServiceConnector component (described in Chater 3 and 4) and our contribution in this layer. It addresses the issues of service and data integration and enables communication between end users and the distributed services at the Service layer.

Note also that users can directly invoke the available services without using the Geoportal layer. This behaviour is common in OGC-based services since users can invoke such services both directly via HTTP queries encoded according to OGC standard specifications (WMS, WPS, etc.) and via applications that hide to some extent the complexity of the underlying HTTP calls.

The *ServiceConnector* collects user queries, encodes them in OGC-standard format, and connects to the corresponding distributed services. Each type of service (discovery, download, etc.) uses different encodings and service interfaces. We have implemented concrete components (vertical boxes connected to the Service Client box) for each service type. Our main contribution is the WPS API that connects the client interface to the WPS (implemented according to the OGC WPS) and it is deployed in the Service Layer.

5.1.2.3 Services and data repositories

Geospatial services occupy the main part of the proposed architecture. We focus on the Processing Services which provide geospatial and non-geospatial processing capacity. Service design principles and the set of geospatial services are described in detail in Section 5.1.3.

Data repositories are out of the scope in this section, but it is worth mentioning that the *ServiceFramework* deals with data repositories. In the proposed solution, users can add and update resources to the framework which hosts the services and repositories.

5.1.3 Design

We focus on the service layer that comprises the service network (using INSPIRE terminology). As described in the introduction section, the main goal of this use case is to demonstrate the proposed architecture and Service-Framework components.

ServiceFramwork deals with increasing the availability of tools by wrapping scientific processes and expose them as distributed processing services according to the OGC WPS specification. Wrapping standard functionality as a standard WPS allows the processes to be reused in multiple scenarios. Providing these services with standard interfaces constitutes one ingredient to achieve GIS interoperability (Díaz et al., 2008).

In this section, we describe the design of the service network processing components independently of any technology, i.e., the externally visible behaviour of the service types (e.g. UML specifications of the interface types of the processing services.)

5.1.3.1 Granularity

Together with the hydrology team, we have split the hydrological models into increasingly smaller pieces in order to identify the relevant processes. This recursive methodology continues until we encounter the desired level of granularity for the given processes. The criterion to stop the top-down decomposition is to consider a given process to be specific yet functional enough to not to be split again. That is, subsequent divisions would make no sense for the specific application requirements. The stop decision stems from a consensus between service designers and hydrology experts. The resulting processes then become candidate processes for implementation as service processes within the geospatial processing services. Suitable basic processes are those which perform a basic function (subject to application requirements) and can be potentially reused in other workflows. The ultimate goal is then to create a library of well-documented, stable geospatial services in which customized and elaborated functions (workflows) rely on other functionality-focused and well-tested services. In this case it makes sense to talk about fine-grained services in order to increase their reusability.

To pursue the maximum reusability, we have exposed the distributed processes to include all the basic tasks that were involved in the scientific workflows. However, it is common to find chains of tasks that are called repeatedly along the workflows, which involve two or more of the basic functions mentioned before. A recurring practice in GIS application development and in SOA in general is to combine elementary operations into more complex tools in order to address specific user requirements. In these cases, repetitive chains of processes have been grouped forming a new process with larger granularity and showing better performance. This strategy decreases development time and gains efficiency by avoiding unnecessary service calls and by minimizing data exchange over the network.

As an example of coarser-grained process, Figure 50 shows a simplified sequence diagram for the Elevation Zones process. The Elevation Zones process contains an integrated chain invoking first the Reclassify process and then the Vectorize process, both processes taken from the Sextante geoprocessing service (See Table 1). SEXTANTE (Olaya, 2007) is a collection of geoprocessing routines and is available as free software. The Reclassify and Vectorize processes expose well-known pieces of functionality and are independently reused in other scientific workflows along the hydrological models, though these processes formed a more coarse-grained service in the given example because the Elevation Zones process was called several times as part of different scientific workflows.

The fact that a given service is reused several times justifies its level of granularity. The more fine-grained a service is, the better its resusability. However, it is always recommended to use coarse-grained services for improving performances, so long as they are somewhat reusable. Both rules hold for the Elevation Zones use case. Furthermore, finding the right balance between service efficiency and reuse is often a subjective matter and depends on the specific application requirements. Therefore, a good practice is to create services at multiple levels of granularity so that one can test the balance between the advantages of fine- and coarse-grained services.

In the given example, the client (WPS API component in the *ServiceConnector* module) interacts with geospatial services independently of the level of granularity. It prepares the requests and processes the results. The Elevation Zones process manages the execution of the contained processes. In particular, the first process called is Reclassify which traverses each DEM cell reading elevation values. According to the desired elevation ranges for the elevation zones, the process then assigns each cell to an elevation zone (500-1000m, 1001-1500m, etc.). Reclassify produces a classified raster file which is fed to the Vectorize process which performs a common format transformation operation, converting the input raster file into the equivalent in vector polygons. The resulting vector file is encoded in GML format and delivered to the Elevation Zones process which sends it to the Processing component in the Application layer.

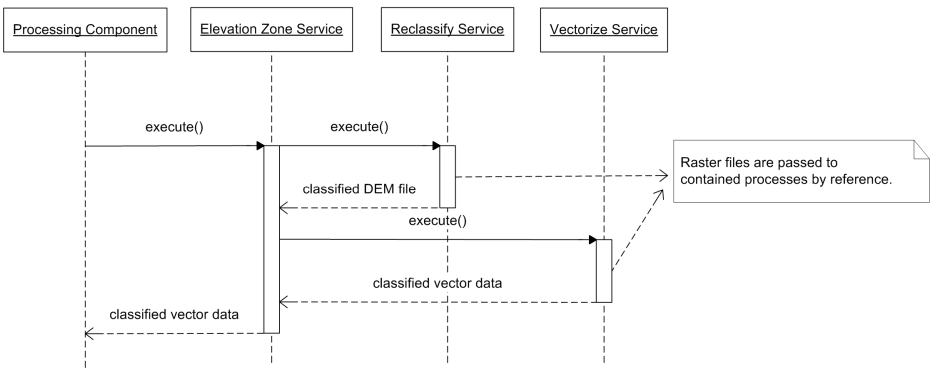

Figure 50. Elevation Zones process workflow

5.1.3.2 Services specifications

As described in Chapter 4, once functionality and granularity has been decided, we must take into account two aspects. First, the potential service types should be identified according to a well-established framework such as INSPIRE. This helps developers organize the spectrum of services since services of the same type normally share the same design process and interfaces. Second, during the implementation phase we choose the most appropriate interface for each service and offer the desired functionality. Figure 51 illustrates these service creation steps.

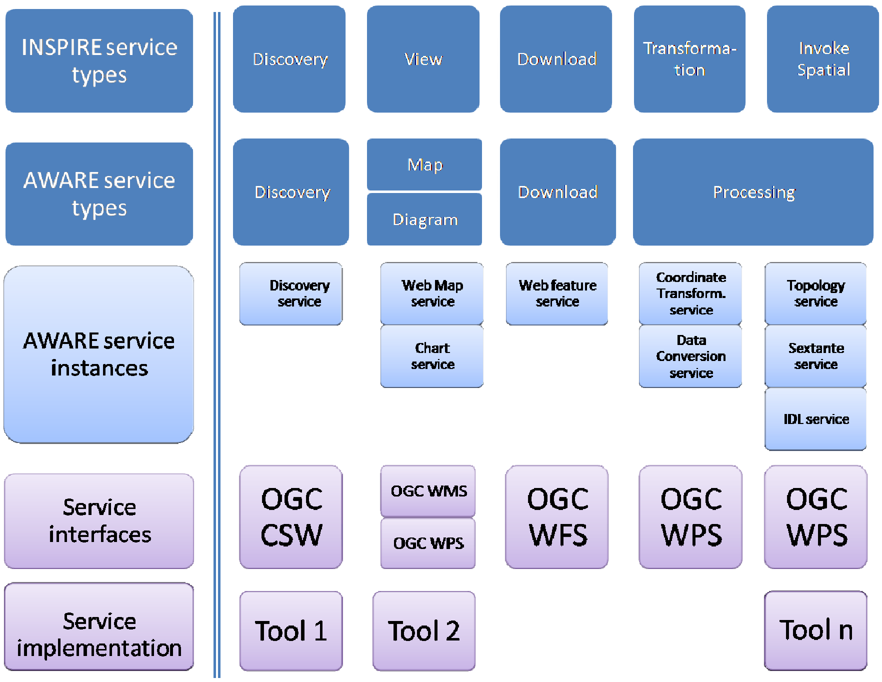

Figure 51. Service perspectives: types, design and implementation. Service types is about INSPIRE vs. AWARE service types. Design is about granularity (instances) and interfaces. Implementation is about specific tools and programming languages.

The row labelled "Service Interfaces" in Figure 51 represents the OGC specifications used for each service instance. It is important to note here that a given application service type may be described by means of various OGC service interfaces and vice versa; the same service interface may be used in various service types. This illustrates that many-to-many relationships between services at the abstract level (type) and specifications at the interface level are possible. For instance, in the use case we described how we offer Web Map and Diagram service types which correspond to the AWARE View services (at the abstract level), but are interfaced with two distinct specifications: one provides maps via WMS interface and the other generates diagrams such as line plots via a WPS interface. The results are identical (images), but with clearly different semantics. Also, the same specification (WPS) may describe many service types. This demonstrates the flexibility of the WPS specification to allow wrapping of nearly any kind of process.

The generated service instances are shown in the third row of Figure 51. The processing service type includes the processing service focus of our work. Table 2 lists the geospatial services and their contained processes. Some of these services are the discovery or download services which, as mentioned earlier, are out of the scope of this work.

We chose the WPS interface to implement the processing services classified under the Processing Service type. Each WPS instance offers a set of processes with similar functionality. Most processes within the Topology geoprocessing service are fine-grained and thus are highly reusable (e.g. Area, Intersection, Buffer, and MaxExtent are concerned mainly with topological relations, geospatial proximity and/or distances between geospatial objects). Other fine-grained processes, however, are rarely reused in other scenarios because they are subject to specific application needs. Processes within the Chart geoprocessing service, for example, are almost entirely devoted to producing line plots and diagrams that are specific to the AWARE context. Other fine-grained services with higher reusability levels include Classify, Vectorize, and Thiessen.

AWARE Service	***Type* / Specification**	**Service processes**	**Description**
Catalogue	*Discovery* / OGC Catalogue Service for Web (CSW)	N/A	It offers the functionality to search and provide all earth observation data catalogued of the study areas in the AWARE project.
Web Map	*View* / OGC Web Map Service (WMS)	N/A	It provides the user with some graphical maps of datasets over the study area.
Chart	*View* / OGC Web Processing Service (WPS)	Depletion Curves Plot Discharge Plot, HBV Runoff Plot, HBV SWE Plot, Sensor Data Chart	It provides diagrams (e.g. line plots) to represent some of the useful information, not as maps, but as descriptive plots showing some information in a

			graphical way.
Web Feature	*Download* / OGC Web Feature Service (WFS)	N/A	It provides users with some vector data (GML) over the study areas.
Coordinate Transformation	*Processing* / OGC Web Processing Service (WPS)	TransCoordGMLPoint, TransCoordPoint, TransCoordPoint7P	It converts coordinates from a source reference system to a target one.
Data Conversion	*Processing* / OGC Web Processing Service (WPS)	Shp2GML	It converts from shapefile format to GML format.
Topology	*Processing* / OGC Web Processing Service (WPS)	Area, Intersection, Buffer, Max Extent, Snow Percentage, Get Feature By Attribute, Thiessen	Topological operations and interpolation algorithms.
Sextante	*Processing* / OGC Web Processing Service (WPS)	Coordinate Elevation, Stations Elevation, Elevation Curves, Elevation Zones, Hypsometric Elevation, Reclassify, Vectorize	Image processing algorithms, raster computations.
IDL	*Processing* / OGC Web Processing Service (WPS)	Snow Interpolation, Calibration, Simulation, K Coefficient Computation	It wraps polynomial interpolations and routines in IDL.

Table 2. Services and processes used in the AWARE Application.

5.1.3.3 Distributed processing specification

This section focuses on the service layer and its processing components. Figure 51 offers an overview of the Network Services defined in AWARE which are derived directly from INSPIRE types. AWARE processing services are placed under the type PROCESSING. Furthermore we have added specific plot and diagram capabilities to the view service type. Both capabilities are different representations of displaying geospatial data and this approach has been followed by other projects like ORCHESTRA[1] and Sensors –Anywhere project (SANY)[2.]

Once the processes needed to perform the functionality to run the hydrological model are identified, they are then grouped into modules with similar functionality. Each module is designed as a web service which provides these processes. Next, a subset of the services shown in Table 2 is described.

1 http://www.eu-orchestra.org
2 http://www.sany-ip.eu/

Annex A provides a full description of the AWARE Processing Services that are available in the AWARE Service network with an abstract specification of the processes provided by each of them: DataConversion Service based in the INSPIRE transformation service; Topology, Sextante and SRMIDL Services which belong to the AWARE Processing service type; and Chart Service which we have classified as ViewChart service type that is based in the INSPIRE View Service type.

5.1.4 Implementation

In this section we describe the implementation of the processing service specifications. From the technology viewpoint we define the platform to be used.

We use OGC standards as they are widely used as standard interfaces for services in SDIs. We wrap scientific algorithms as web processing services using OGC standards specifications in order to place them in an SDI so they can be accessed by different users in different scenarios.

When analyzing the functionality needed to execute the hydrological model, we propose to wrap and reuse the functionality in FOSS projects and expose them as web processing services.

5.1.4.1 Distributed processing services: OGC Web Processing Service implementation specification

Annex B describes the AWARE Processing services that have been implemented according to the OGC WPS implementation specification.

5.1.5 AWARE Hydrological models execution

Figure 52 shows an example of the AWARE Application running the first step of the one hydrological model. The Geoportal for the AWARE application integrates a Catalogue service client, which connects to the AWARE Discovery Service offering the discovery capability and providing the user the available data. The right side of Figure 52 shows the map viewer displaying the basin boundary together with the location of meteorological sensors.

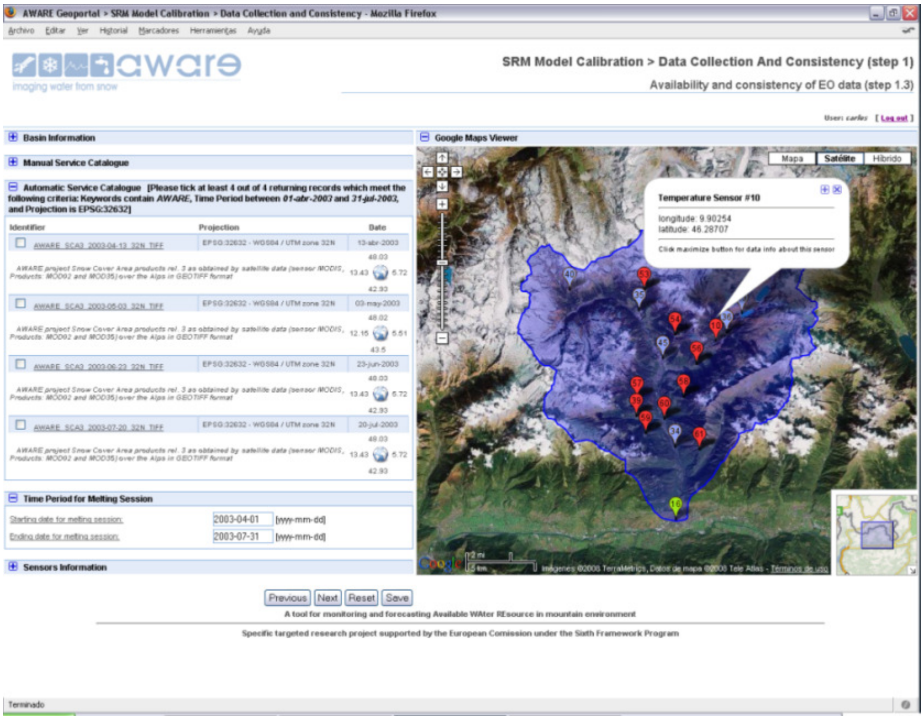

Figure 52. User interface for retrieving metadata records from a catalogue service.

Figure 53 shows an example of the possibilities offered by the AWARE Application for data visualization and exploration. Users may click on a sensor icon to obtain more information about the sensor data. This action is executed on the server side via requests to the Chart WPS.

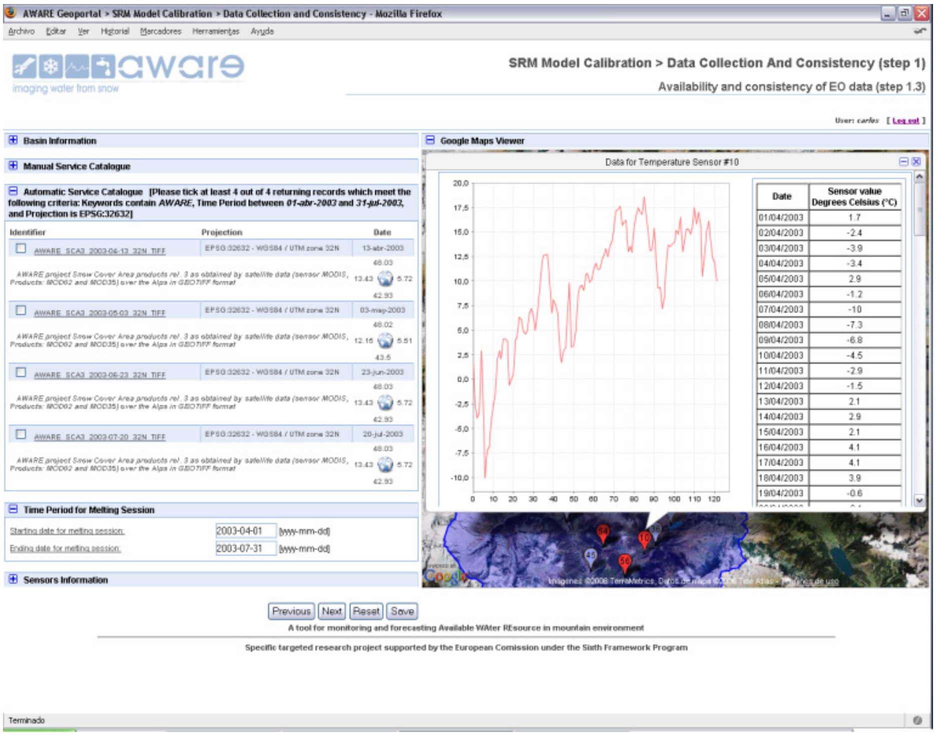

Figure 53. User interface to visually inspect temperature sensor data

Figure 54 shows the activity diagram of the workflow that calculates the elevation zones of a watershed. It is important to note the disparate datasets involved in this workflow: the watershed DEM in raster format, the DEM projection given as EPSG code, a range of pairs (min-max) of elevations to define each elevation zone, and the watershed boundary in GML format. Manually analyzing these disparate inputs together with the intermediate datasets is a tedious task, and so this is often performed inefficiently.

The hydrological model Logic component (see Figure 49) implements the scientific workflow orchestrating the geoprocessing services in the Service layer. The first task (i) calculates the elevation zones given a DEM file, its projection and a range of pairs (min-max) of elevations. This task is performed by making a request to the *ElevationZones* process provided by SextanteWPS through the WPS API component in the *ServiceConnector*. The *ElevationZones* process calls two raster analysis processes within the SextanteWPS. It returns a GML-encoded output with the elevation zones. As the DEM's extent may be different than the watershed, it is necessary to perform an intersection between the elevation zones and the basin boundary. The next task (ii) carries out this by sending a request to the *Intersection* process provided by the TopologyWPS geoprocessing service and returns the intersected area. Once delimited, each elevation zone is extracted from the elevation zones file according to an attribute (representing category (iii)) by executing the *getFeatureById* process provided by the Topology geoprocessing service.

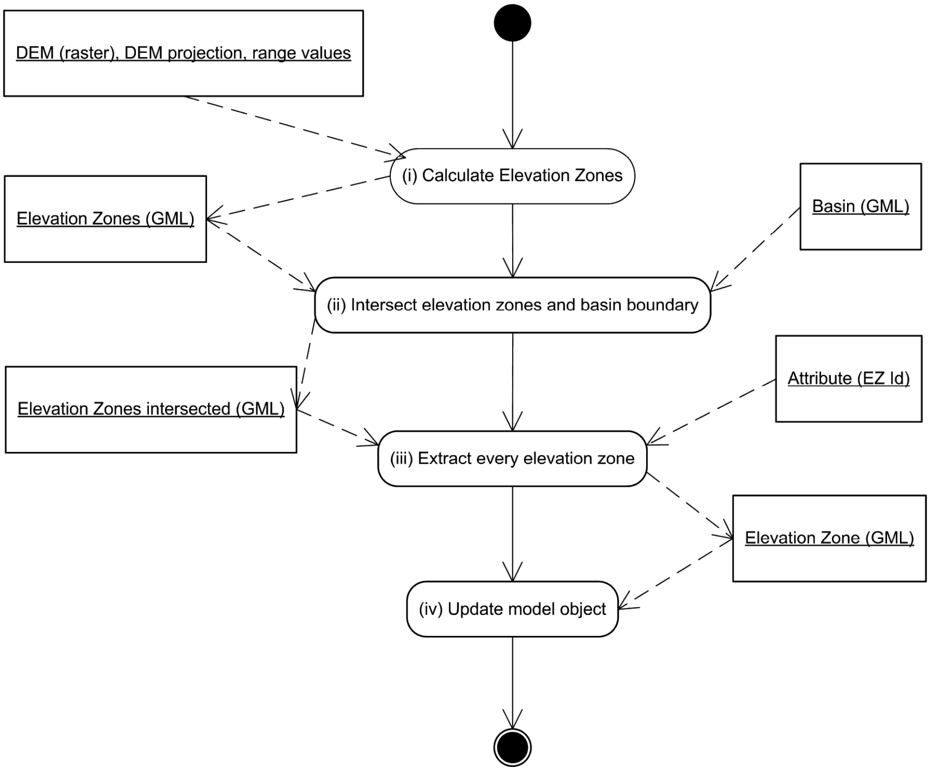

Figure 54. UML activity diagram to calculate the elevation zones

Figure 55 shows, on the right side, the elevation zones created by the *ElevationZones* process in SextanteWPS. The left side shows the values of the elevations of the meteorological stations that were calculated by the *StationsElevation* process in SextanteWPS.

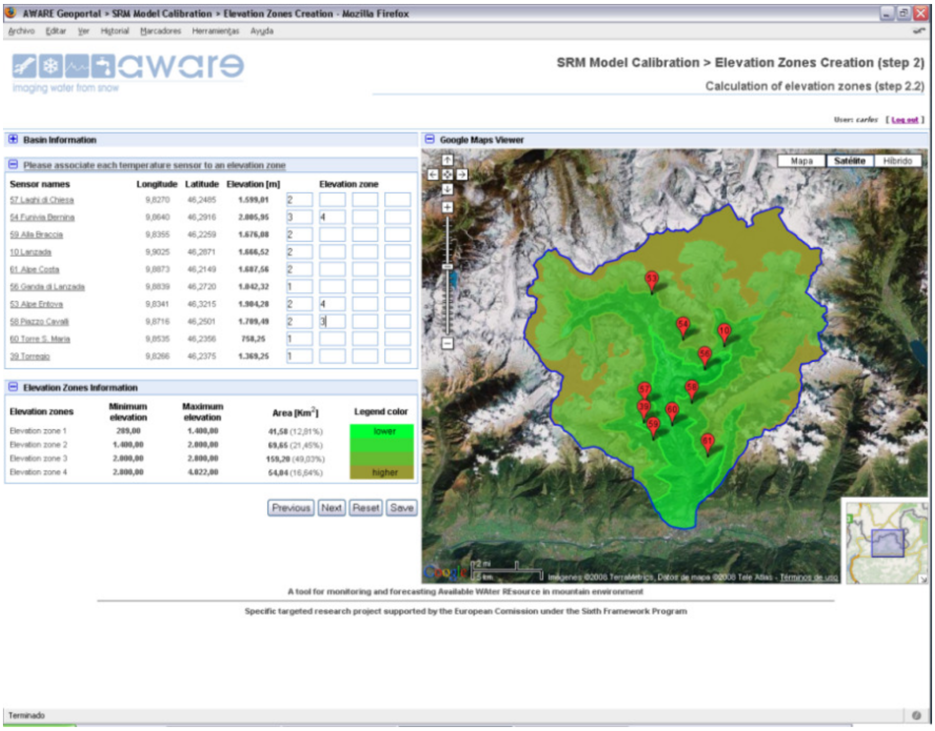

Figure 55. User interface displaying the elevation curves for the watershed

Figure 56 shows the results for the hydrological model calibration. The results are presented in the form of line plots to facilitate the interpretation of the model. The top line plot compares the calibrated and measured runoff discharge of a watershed, whereas the line plot just below displays the measured against the simulated runoff discharge of the same watershed.

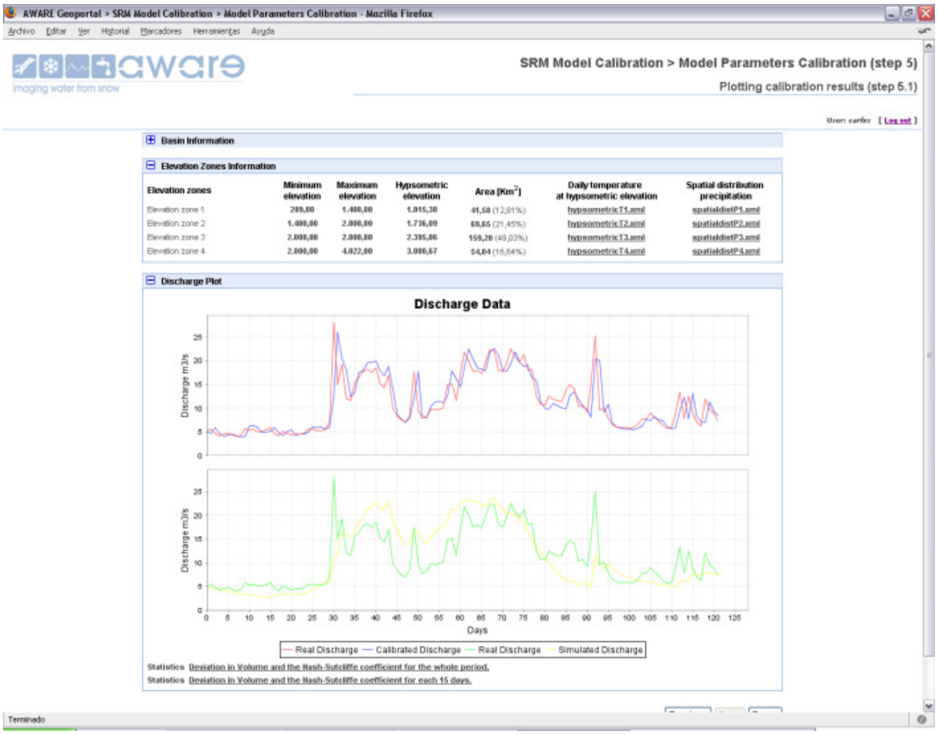

Figure 56. Results with plots comparing calibrated, real and simulated discharges of the watershed.

This use case has demonstrated that the proposed architecture and its components can improve the linking of processing capabilities within the GII. This scenario and the proposed solution demonstrated that the GII can be a useful platform to fulfil advanced requirements like processing and modelling.

5.2 Experiments on the EuroGEOSS project: Improving data availability to perform a protected areas damage assessment.

Forest fire disasters are increasingly becoming frequent events around the globe. Forest fires analysis and the strategic planning in combating them require the development of GII, data and services interacting with each other.

In this context, the EuroGEOSS project pursues the improvement and establishment of interconnection among systems and data structures about forests as well as identification of options and interfaces to benefit from resources available at different levels. EuroGEOSS demonstrates the added value by making existing systems and applications interoperable and used within the GEOSS and INSPIRE frameworks. EuroGEOSS focuses on the application areas and the multidisciplinary interoperability aspects to opening them up, linking them, and making them GEOSS components.

As described in Chapter 4, our purpose is to help and assist users to build these systems on top of these policies and deploy all the standard components to share the required resources. We propose to evaluate the ServiceFramework module into the EuroGEOSS systems to let users add and integrate data resources. The premise is that the use of the ServiceFramework could help to increase the availability and visibility of forestry resources in the EuroGEOSS systems in an interoperable way at global, regional and local levels.

5.2.1 Analysis

From the three Social Benefit Areas (SBA) addressed in EuroGEOSS – forestry, drought and biodiversity – we focus our scenario on the forestry area. The European Forest Data Center[xxxiv] (EFDAC) is being implemented in compliance with the guidelines of INSPIRE. The EFDAC is built on the basis of existing systems, such as the European Forest Fire Information System (EFFIS). Therefore the integration of existing datasets, tools and applications ensuring the overall system's interoperability is the main challenge and goal for the further development of EFDAC (EuroGEOSS WP3, 2009).

An overview of the current EuroGEOSS systems such as EFDAC was performed and we discovered a common gap across these systems. Although there are many forestry datasets available, not all of them have been published via catalogue systems nor are they available via standard OGC web data services (EuroGEOSS WP3, 2009).

We have chosen EFFIS as the working framework in which to validate our premise and evaluate the proposed approach. The core of EFFIS consists of a scientific and technical infrastructure operating a web-based platform.

Next we describe a use case in which we assist the experts on forest fires in updating the content of the system as a step integrated in their daily workflow. As a consequence, by letting users add massively data resources as interoperable services, we address the gap of the lack of resources available through standard services.

At the moment, the web-based system permits experts to access the forestry data at the European scale which can then be used to produce reports. As it has been described in the previous chapter, experts can be provided with distributed and interoperable tools to process these data.

There are two points to be demonstrated by using the ServiceFramework. First, users at all administrative levels can be assisted to add local datasets as standard services, so they can be accessed in an interoperable fashion by the rest of EFFIS users. Second, we demonstrate how to assist experts in updating the GII. ServiceFramework allows users to generate new information connecting to available WPS. Furthermore it deploys this generated information to the standard data services to be shared with the rest of the users at all scales. This offers a more advanced way to share resources since they can be massively added to the European SDI to be made available in an interoperable fashion.

5.2.1.1 Scenario

Our approach adds a new role to the user. Now, GEOSS users can participate in the maintenance and updating of the GII. This means that a GEOSS user could deploy newly generated resources as interoperable components of the GII after searching, accessing and analysing data. Improvement of interoperability with global, regional and local datasets and services related to forest fires could increase the effectiveness of EFFIS.

Next, we describe the events of a particular scenario selected from the EuroGEOSS project to demonstrating whether our proposal fits the requirements. This scenario illustrates a scientist's workflow and it allows us to identify where the ServiceFramework makes sense to help its purposes:

- National forest experts through the web interface **query** (WMS/WFS/WCS) the TREES database for images and other forestry-related datasets and **validate the** TREES database using the queried results.
- Forest/vegetation experts through the web interface query (WMS/WFS/WCS) other global datasets (WCMC, OFAC, FIRMS) and **update** the protected areas database in conjunction with the queried results.
- Forest experts **analyze**, **correct and update** maps of land cover around and within the protected area in the database.
- Using the web data services (CS-W/WMS/WFS/WCS), end-users **search** and **access** forestry information from national and local levels in Spain.
- Forest fire experts or GEOSS users **search** and **access** forest fire layers that are available in EFFIS (e.g. information on forest fire danger forecast, forest fire damages, maps of burned biomass, atmospheric emissions).
- Forest fire experts search (CSW/WMS/WFS) and **select** through a catalogue the forest fire thematic layers, and use them for **analysis** and research purposes within the Map Viewer (e.g. calculation of burnt area, evaluation of accuracy of the results according to different data sources). The **combination** of forest fire layers with other forestry thematic layers

could be used to **evaluate** and **analyze** the impact of forest fires to protected areas. We will enumerate the steps carried out when **processing**, for instance, the calculation of burned areas:

- EFFIS connects daily to the NASA database via FTP and downloads MODIS data including land surface and thermal anomalies which are stored locally. These raw data are processed using desktop applications including the next methods.
- Calculate a polygon (buffer process) of one square kilometre around the thermal anomalies or hotspots.
- Filter the hotspots using the CORINE land cover. If the hotspot is not on a natural land (e.g. agricultural or artificial), the hotspot is removed.
- Filter the hotspots using multitemporal analysis. In this step some time series are compared by visual classification.
- Manually digitalize the polygons with burned areas.
- Run statistics and generate a local file with the digitalized polygon.
- This data is shared via email to the national forestry agencies and deployed periodically to an internal OGC WMS by the skilled person in the department.

• The end–user who would be combining the forest fire thematic layers from different sources **produces the additional layers and exports/saves them locally**.

The described scenario shows how in the traditional scientist's workflow the GIIs are currently used to discover and access resources. The major problem seems to occur when, adding or updating newly generated resources because they are persisted only locally and shared by mail or via FTP. Among others, the complexity of deployment mechanisms are one of the reasons why newly generated resources created daily by experts are not being persisted in the GII. This leads to a lack of available resources in an interoperable way and an inefficient maintenance of the GII.

5.2.1.2 Use case: Protected areas damage assessment

Now we describe a use case to integrate and evaluate the ServiceFramework. The idea is to use the ServiceFramework to provide an alternative to the current scientific workflow. We propose to assist users to generate data and extract information to later be persisted in the GII. In our use case a user performs a protected areas damage assessment. It consists in overlapping the burned areas in a region with the protected areas to estimate the damage of these important biodiversity areas.

In our approach we try to provide a more participative environment where our proposed architecture and the ServiceFramework allow scientists not only to access resources but also to generate new information and to deploy it in the GII. This workflow is demonstrated in Figure 57 and the steps are described below.

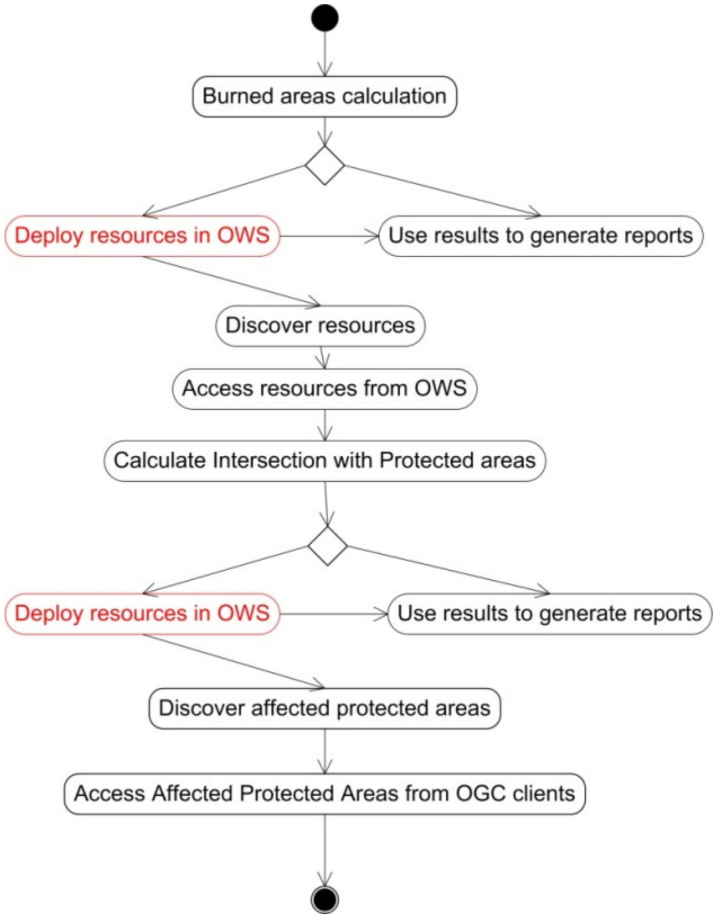

Figure 57. Activity diagram with the tasks of the workflow of the selected use case

- After executing the process of calculating the burned areas, the user could use the information to generate reports and, by means of the Service-Framework, could also to deploy it as a data service in the GII.
- To proceed with the assessment, the expert searches for and accesses layers of burned areas and protected areas. This task can be relatively easy

if they have been previously deployed in the data services and published in Catalogues.

- To calculate the intersection of these two datasets, the user should own the software that performs the function or the user can search for a WPS integrated in the GII offering this functionality. At this point we would like to emphasize the added value of deploying spatial functionality as distributed WPS, because for this use case, we could easily reuse the WPS that was generated and deployed from the previous AWARE project described in Section 5.1. This demonstrates the advantage of WPS reusability.
- The user accesses and runs the WPS containing the buffer and intersection processes using the ServiceConnector component.
- The user gets the processing result of the intersection between the protected and burned areas.
- Using the ServiceFramework*,* the user can preserve the generated information by deploying it as a new dataset (i.e. "affected protected areas") as part of a WMS and WFS within the GII.

5.2.2 ServiceFramework integration in EuroGEOSS architecture

The multi-disciplinary environment, which characterizes the EuroGEOSS capacity, requires the support of different data models and protocols (i.e. standards). This is a challenge for the clients which must implement many different interoperability protocols and data models. (EuroGEOSS WP2, 2010). To address this issue, an extended SOA approach can be used to provide a harmonized access service: the SOA-brokering approach. This EuroGEOSS approach is shown in Figure 58 where we can see the EuroGEOSS Initial Operating Capacity (IOC). Data and service providers can be accessed through the Catalog and Broker component which maximizes the interoperability to those accessing for instance from the GEOSS Common Infrastructure (GCI).

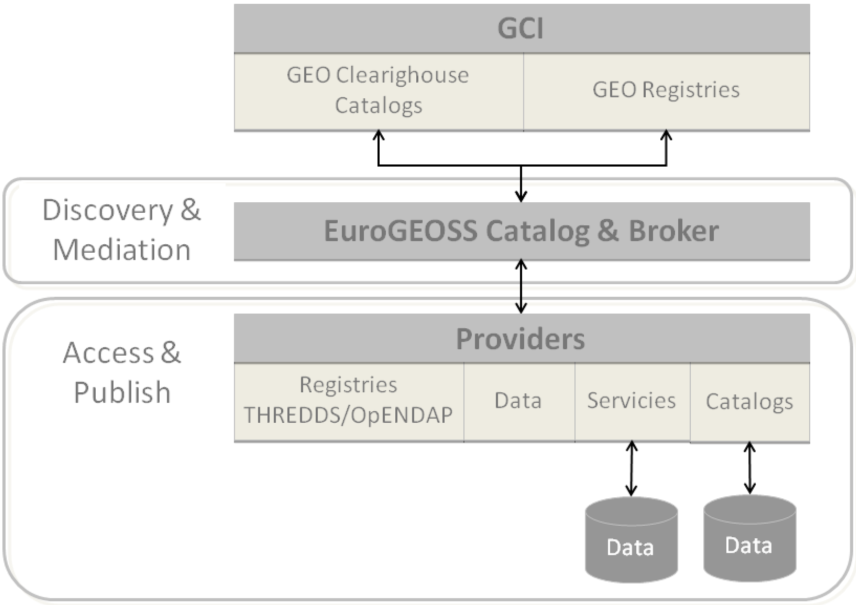

Figure 58. EuroGEOSS IOC

The integration of the ServiceFramework in this architecture will consist in assisting users to deploy resources as services as we can see in the Providers layer, and to catalogue these resources in the Providers Catalogues to be later accessible by the Broker component. The EuroGEOSS brokering component implements a framework to federate well-accepted catalogue, inventory and access standard services (EuroGEOSS WP2, 2010) as we can see in the Figure 59 (EuroGEOSS, 2010).

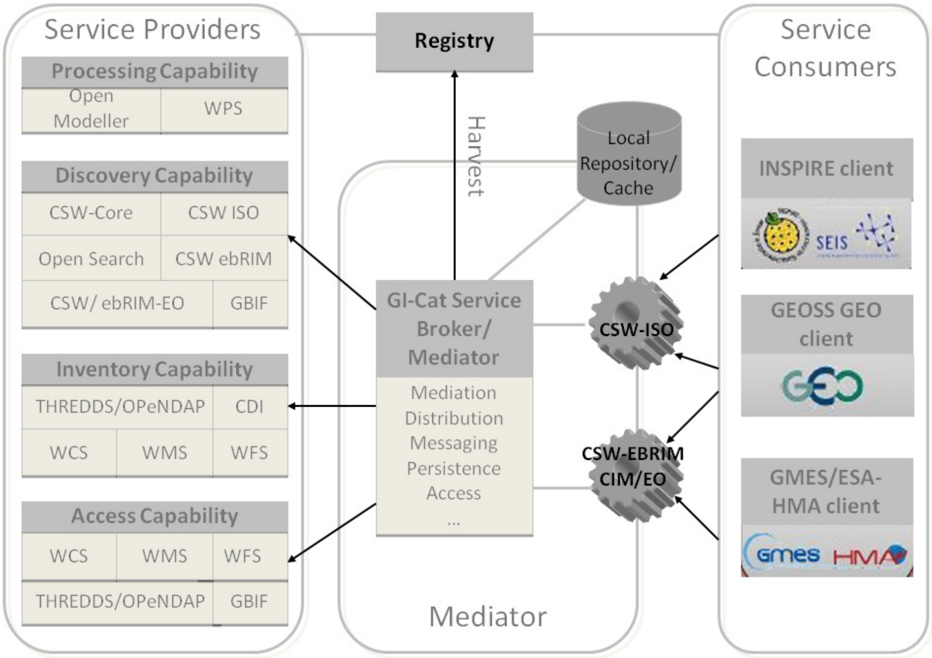

Figure 59. Details on EuroGEOSS Broker component

To illustrate the ServiceFramework integration in our use case the Figure 60 depicts the conceptual view for the integration of the EuroGEOSS Forestry component in the GEOSS Initial Operating Capacity (IOC) (EuroGEOSS WP3, 2010)

For the EuroGEOSS Forestry IOC a new application and interface layer system shall be created and shall centralise the Forest metadata catalogue server and query/view GUIs. A unified Map Viewer GUI component together with a harvestable CSW ISO AP metadata interface (INSPIRE Discovery Service) endpoint and data mapping interface endpoints, i.e. WMS, WFS and WCS (INSPIRE View and Download services) for system to system geospatial resources exchange. The interface endpoints (CWS, WMS, WFS, WCS) shall be used by the FIOC clients both inside and outside the forest systems, i.e. GEOSS and ad hoc Map Views and GEOSS Clearinghouse and ad hoc catalogue services (EuroGEOSS WP3, 2010).

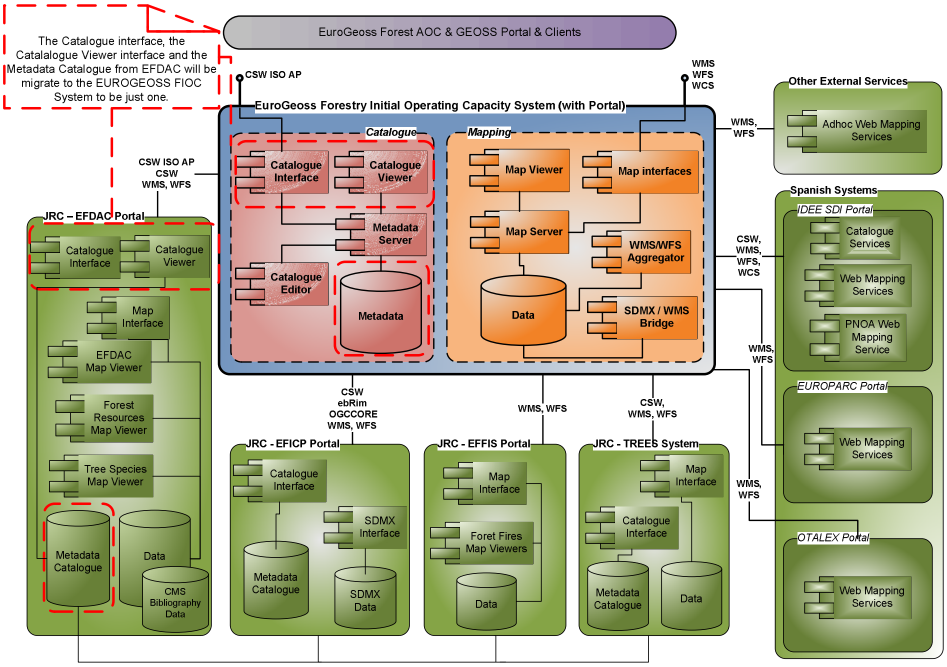

Figure 60. Proposed Architectural Overview for Forest Capability (blue)

Therefore, in this context, the ServiceFramework can assist users to provide, to the forestry systems, resources as standard components as services to be reused throughout this service-oriented architecture. Each sub-system can offer a portal as an entry point that contains the application logic and domain-specific tools as we have pointed out in the proposed architecture in Chapter 3.

Zooming into the details of EFFIS, Figure 61 shows how the *ServiceFramework* could be related to the architecture of this forestry system. As described in Chapter 3, the *ServiceFramework* layers correspond to the layers defined in the GII and is in compliance with the INSPIRE directive.

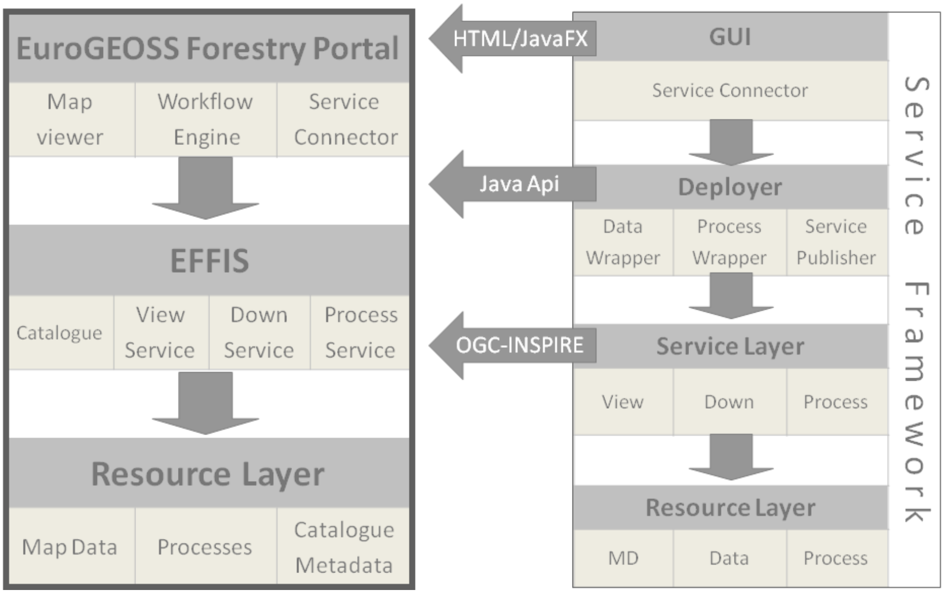

Figure 61. EFFIS architecture with the relations with *ServiceFramework*

The *ServiceFramework* module functionality could be integrated into any of these portals at the application level. The *ServiceFramework* interface is implemented with JavaFX technology, and as we describe later, it can be integrated according to JavaFX characteristics. On the other hand, the *Service-Framework* functionality could be integrated at the Deployer level where functionality can be accessed through its Java API.

5.2.3 Implementation

This section is devoted to describing the details of the *ServiceFramework* implementation related to this use case.

5.2.3.1 User Interface

The user interface of the *ServiceFramework* has been developed using JavaFX technology. Figure 62 illustrates the UML class diagram where we can highlight the view, *edu.uji.serviceframework.view,* and the controller, *edu.uji.serviceframework.controlle*.

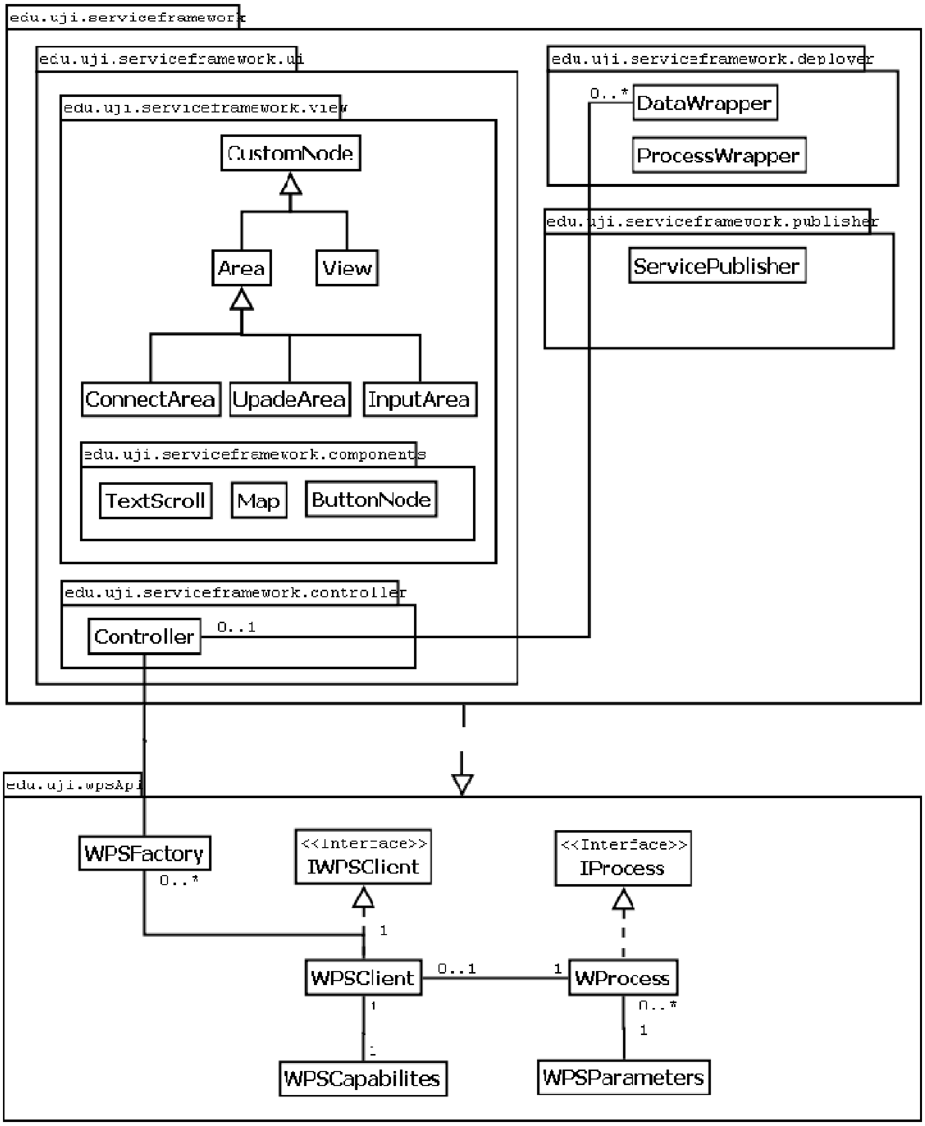

Figure 62. Service Framework Simplified UML diagram

The JavaFX user interface development is based on the classical concept of scene graph for the implementation of graphical applications (Weaver et al., 2009). Following this scheme, *edu.uji.serviceframework.view* implements the *View* as the entry point to the user interface, the *Area* to simplify the definition of graphic elements, and the *Components* package with helper classes that allow Java components to incorporate into the scene graph.

Likewise, the *edu.uji.serviceframework.controller* package is developed following the *singleton* pattern (Freeman et al., 2004) where any instance accesses the same content and execution. JavaFX implements a binding system on the variables to interact between classes within the View and Controller logical layers. For instance, *Controller* variables may automatically trigger processes to update the graphical elements in the View.

The Controller delegates user request to the adequate modules, forwarding the control to the WPS API component to connect to WPS and to the Service Deployer node to wrap resources as web services.

5.2.3.2 User-driven data integration strategy

Figure 42 illustrated the steps and components involved in the ServiceFramework strategy to assist users in deploying data resources in the information infrastructure as interoperable services. See Figure 42 and its section for more information about the user-driven data integration, where is the user whom by means of the ServiceFramework can process data extract information and integrate it in the GII.

5.2.4 EuroGEOSS Protected areas damage assessment

To illustrate this practical approach, we show the latter sequence diagram (See Figure 57) through the execution of this use case by using the ServiceFramework. The use case involves deriving an estimate of the damage of the protected areas after the forest fires in the summer of 2005 in the Valencian region located on the east coast of Spain. The interactions between the user and the architecture components are illustrated in the following figures.

First, if the protected areas data are not available in the geospatial information system, the user can decide to upload it on the system for future use by other stakeholders. To do so, the user can upload a file directly on the application as shown in Figure 63.

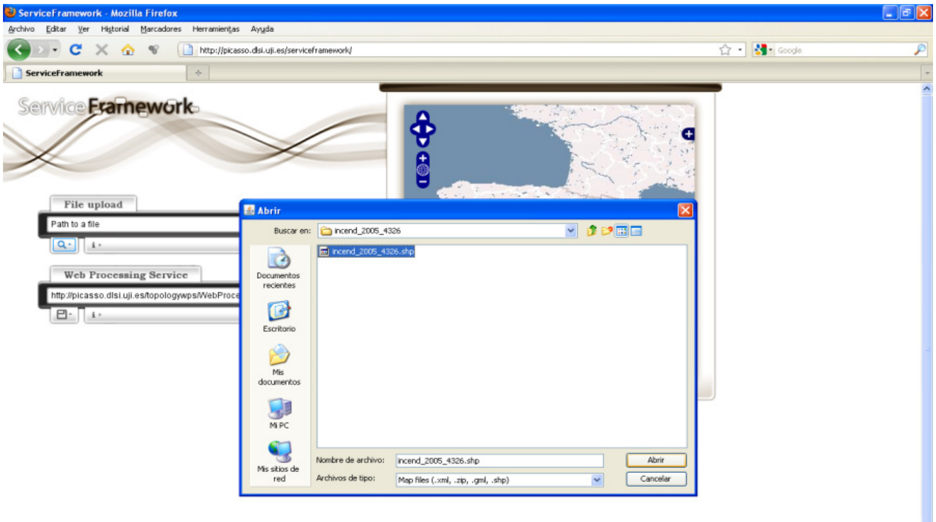

Figure 63. User can upload local data to be integrated in the system.

The uploaded file is displayed on the viewer map. Figure 64 shows how the interface also lets the user select a unique solid colour to render and visualize the dataset.

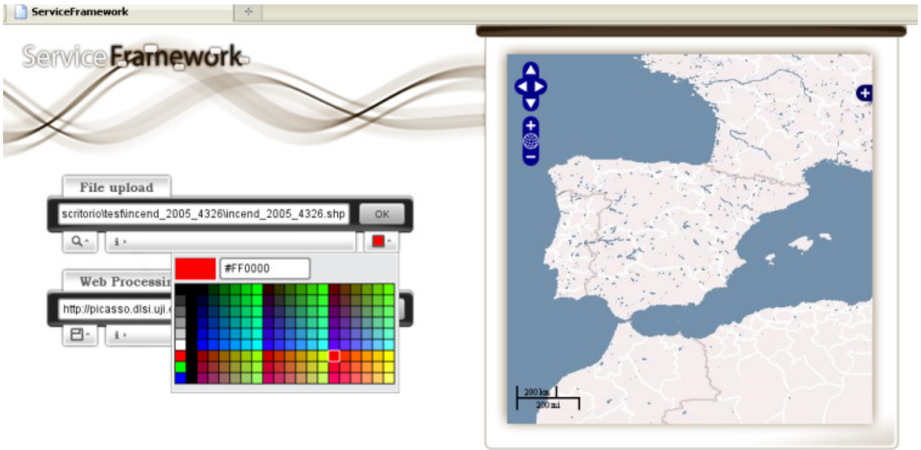

Figure 64. Visualization of datasets.

The protected areas and the burned areas in 2005 are loaded by the user and displayed on the map viewer as shown in Figure 65 Figure 66.

Figure 65. Visualization of datasets (II)

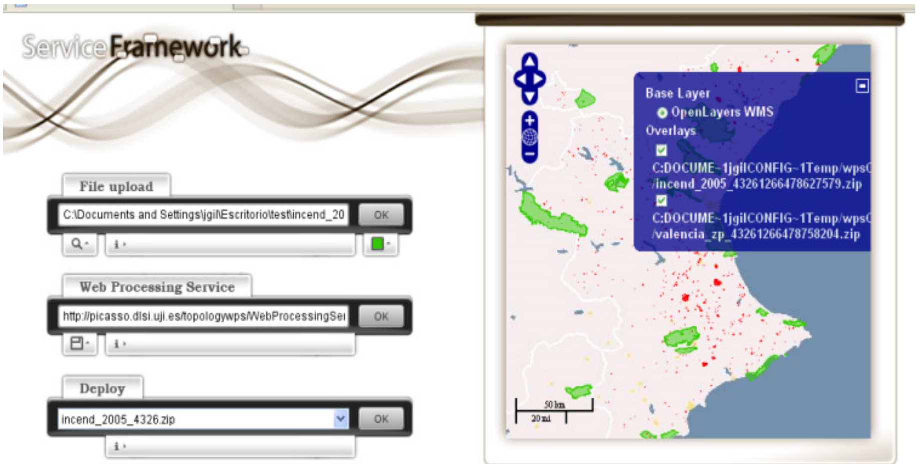

Figure 66. After uploading data, these are visualized on the viewer.

The user is guided through the user interface to deploy his/her resources. At the bottom side of the interface, the user finds the deployment section (Figure 66) where the desired dataset can be selected to be deployed onto the system so next time this dataset can be accessed through a spatial data service.

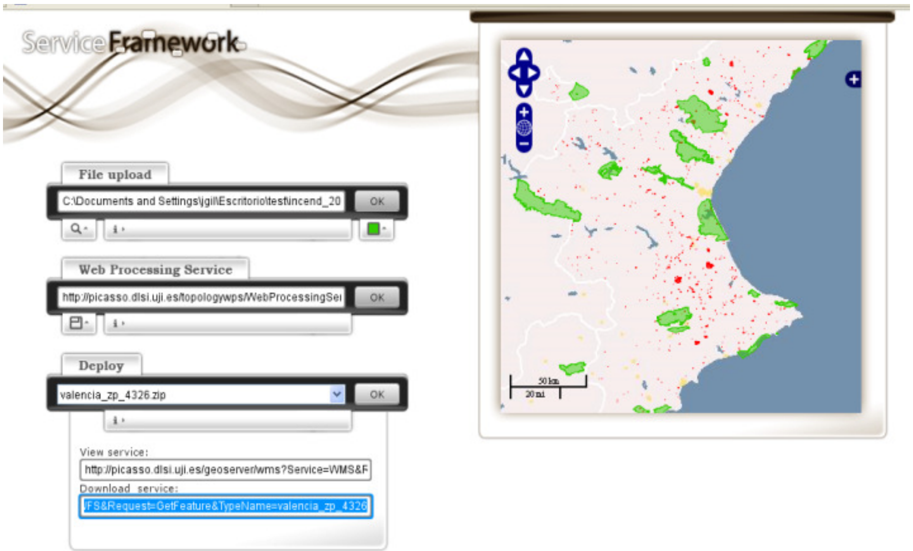

Figure 67. Protected areas dataset deployed as INSPIRE View and download Service type according to OGC standards

Figure 67 and Figure 68 show how the *ServiceFramework* informs the user of the successful deployment of the datasets by showing the information that the data wrapper sends regarding the online access to the datasets by means of an INSPIRE View Service and an INSPIRE Download Service.

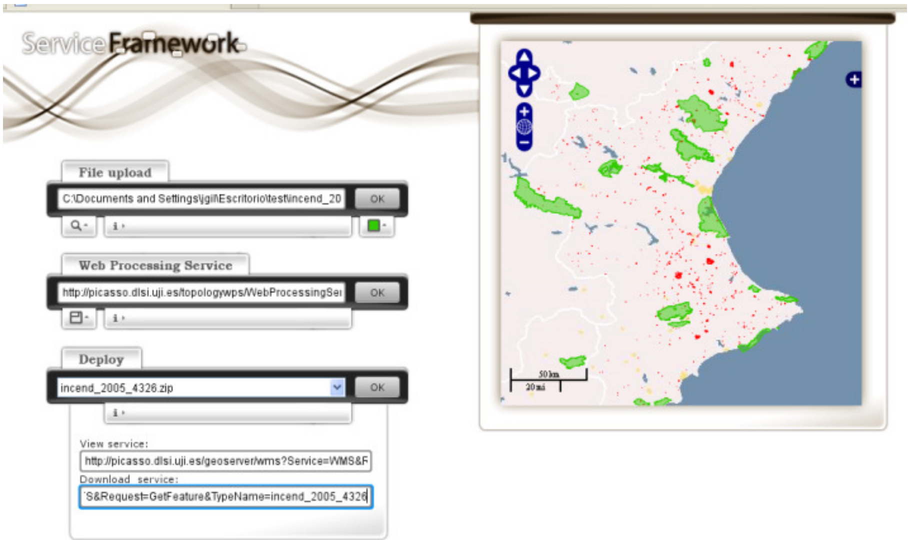

Figure 68. Burned areas dataset deployed as INSPIRE View and download Service type according to OGC standards (II)

Figure 69 shows Geoserver as the chosen implementation of the INSPIRE data services. Our architecture now holds the datasets and serves them publicly and through the standard interfaces to the open community.

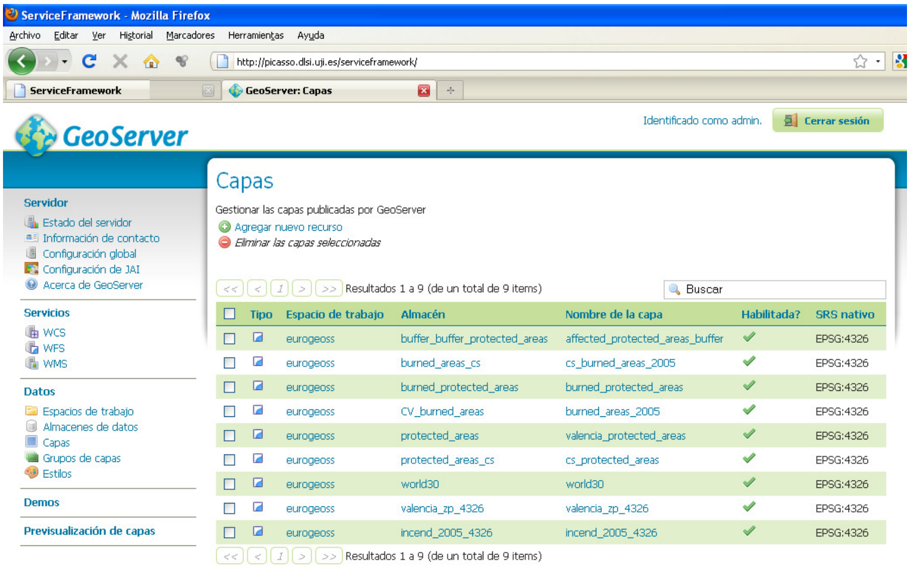

Figure 69. Geoserver as OGC WMS-WFS-WCS implementation serving the uploaded datasets

Besides deploying or publishing local data, the user can extract information by executing a distributed WPS. Figure 70 shows the steps we described previously in Figure 42, where the user can select a given WPS through the WPS API component to retrieve the available processes on it that are displayed in the interface. In this particular case, the user chooses the Intersection process to calculate the protected areas damaged by the forest fires.

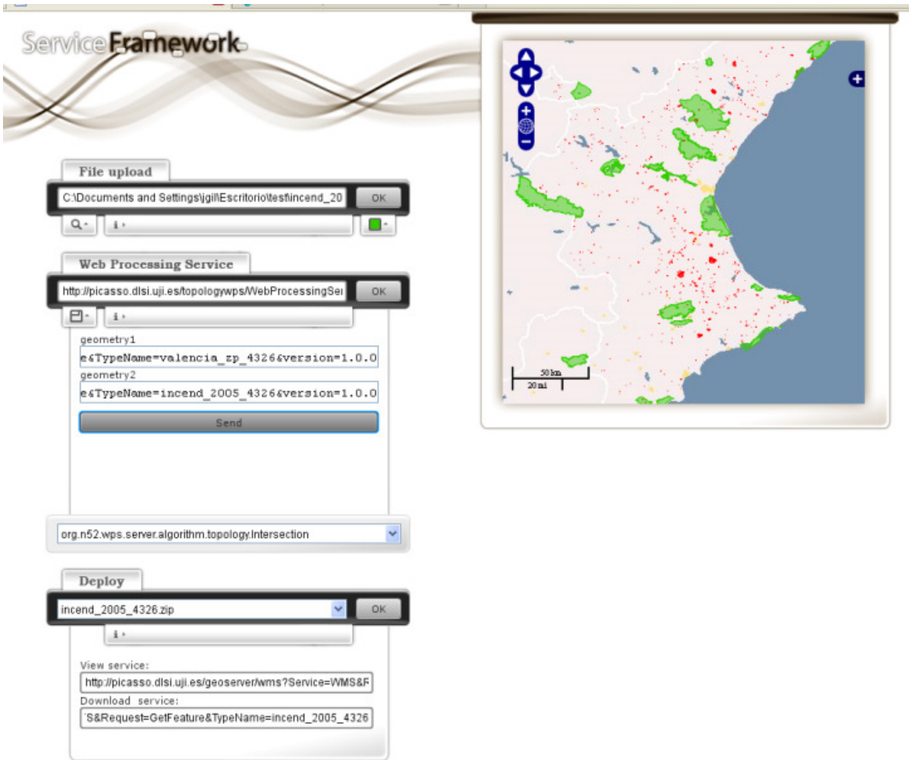

Figure 70. ServiceFramework user interface lets the user run WPS by means of the ServiceConnector-WPS API

To execute the Intersection process, the user is requested to provide two inputs. In Figure 70 we can see how the input can now be a reference to the data service where the user has published the data. After executing the intersection process, the generated information can be visualized on the map as shown in Figure 71.

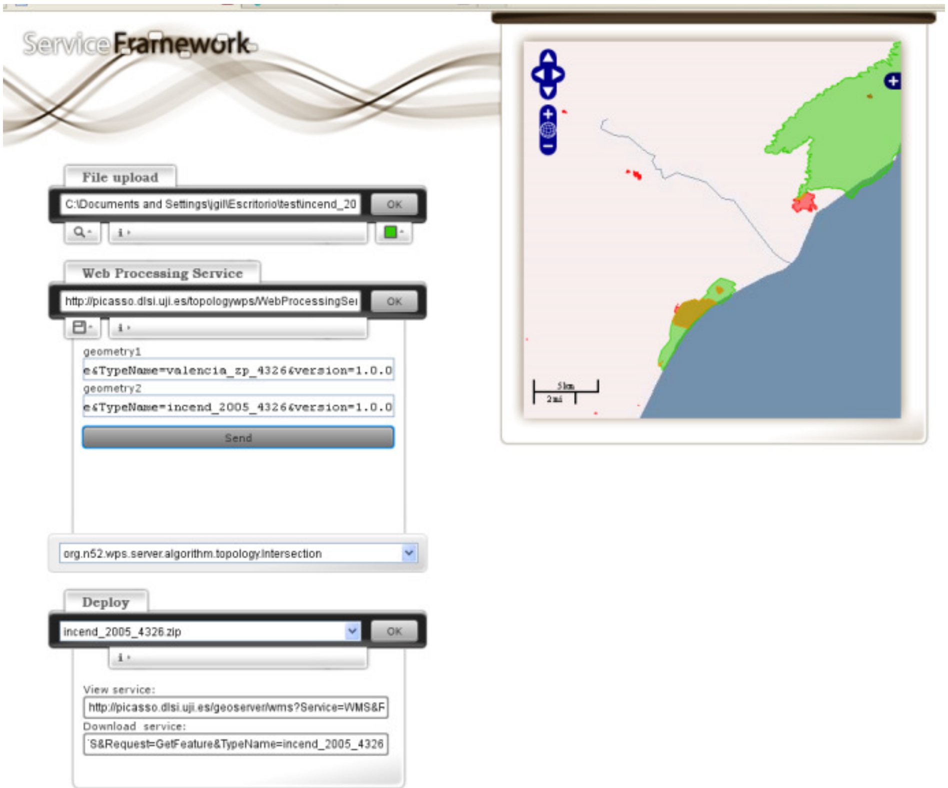

Figure 71. Visualization of process results

One of the contributions of this work is the fact that, after providing the user with reusable distributed tools which can be easily accessed and executed by the *ServiceFramework*, the user is now assisted to integrate this newly generated information in the GII. As with the local dataset, the user can deploy and publish this new information in a data service by just clicking on the deployment section as shown in Figure 72. This improves the SDI maintenance since we provide rapid ways to deliver up-to-date resources to the SDI.

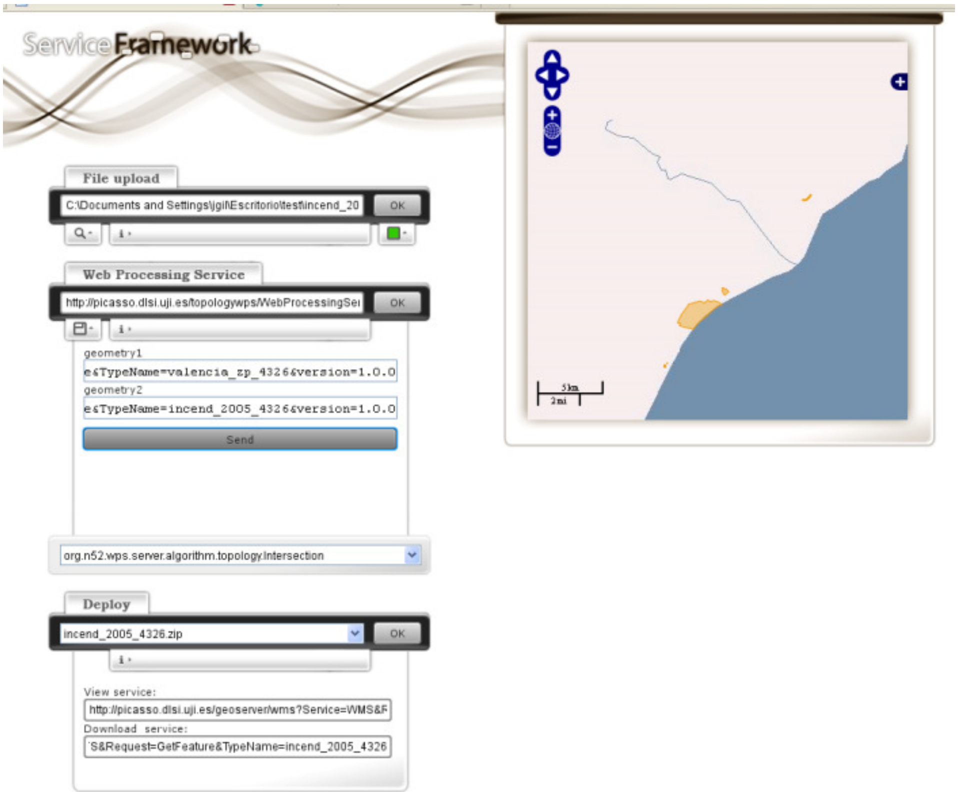

Figure 72. Deploying Processing results as INSPIRE Service types

The last tool to describe in this use case is the ServicePublisher. This is also a proof of concept and aims at publishing services in Open Catalogues Service to increase their visibility. As we can see in Figure 73, the ServicePublisher interface allows the user to select one of the services: WMS, WFS, or WPS, as we described in section 4.3.

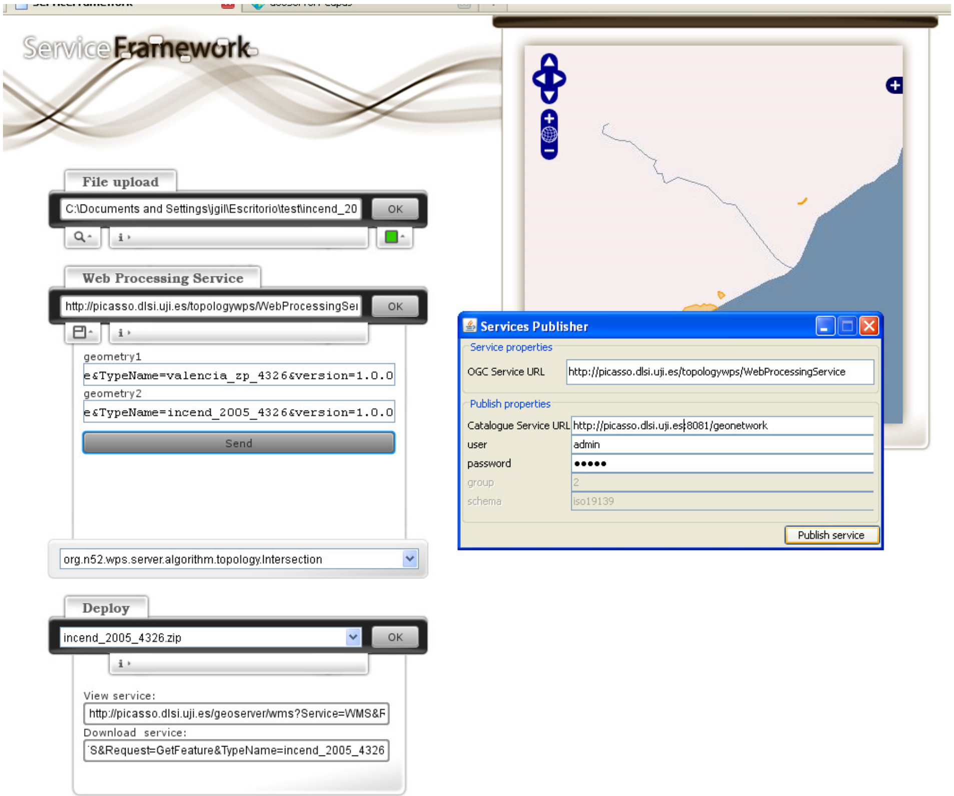

Figure 73. ServicePublisher within the ServiceFramework lets the user publish WPS not yet available in OpenCatalogues

The data and information deployed as standard data services have been integrated generically in the information system. This kind of data management lets other stakeholders access these data from other GIS systems and software (Figure 74).

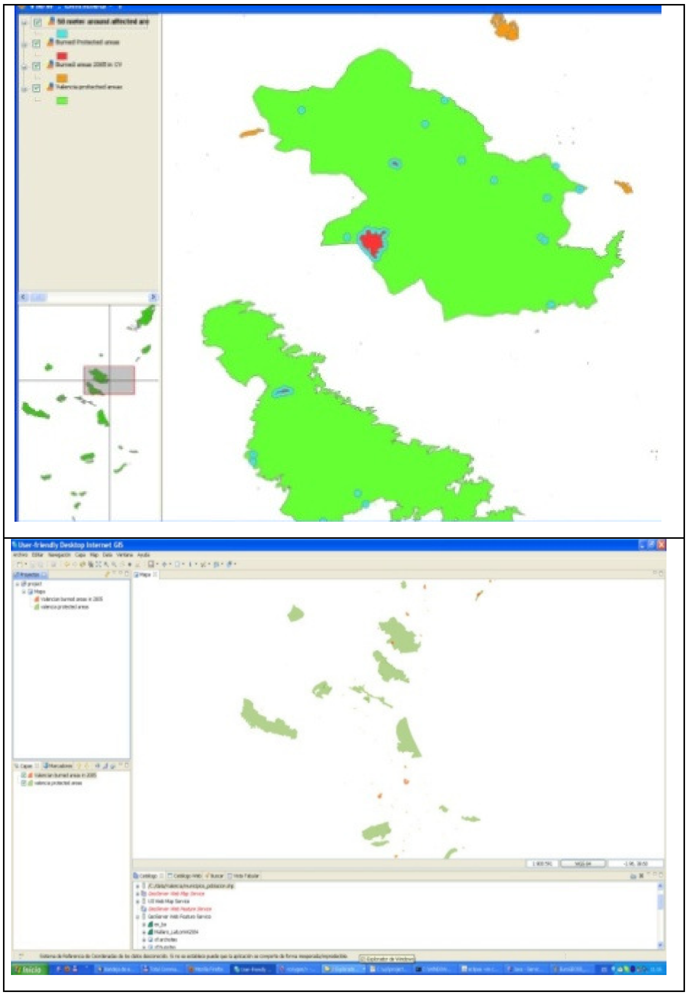

Figure 74. Data retrieval using gvSIG and UDig.

6. Conclusions

This dissertation examines key issues of interoperability of geospatial resources within the environmental domain, reviews GII based on standard web services, and discusses some of the difficulties SDI stakeholders experience in their search for the appropriate tools to perform analysis and distribute resources as standardized components in the GII.

This section summarizes the contributions achieved in this thesis but also limitations in the three research areas described throughout this thesis: availability of processing resources and data resources, along with the increase of visibility of resources in GII.

6.1 Processing resources availability

Certain types of computing applications demand a distributed approach for multiple reasons such as efficiency, reliability and access to data. In our context, expert users not only try to find and access large quantities of data remotely using on-line services, but they also have the need to process them to extract information that helps them make decisions.

Currently, multiple initiatives aim to deploy GII that help users access data in an interoperable way. However, there is a lack of distributed processing, meaning that many processes must be run locally with the need to maintain multiple software products locally. In this work we have proposed and described an approach to provide expert users with remote tools by wrapping scientific processes as distributed standardized web services so that they can be shared and reused in multiple scenarios.

A first observation derived from the scenario experience, which coincides with conclusions in previously referenced work, is that the approach based on distributed geoprocessing services leads to a collection of reusable geoprocessing services, available for other users in the case that they are well-documented and registered in open catalogues. This is possible in principle because WPS-based geoprocessing services do not work with pre-established datasets, but rather they preserve a loosely-coupled relationship between *ad hoc* data and processing capabilities making it possible to chain them to other geospatial web services. We demonstrated a collection of reusable geoprocessing services, available for other users in other scientific domains.

The OGC WPS services have been tested in different contexts (Friis-Christensen et al., 2007; Foerster and Shäffer, 2007, Díaz et al., 2008b, Granell et al, 2010), illustrating that it is possible to combine several geospatial processing services for accessing, processing and visualizing data within GII.

OGC standards seem to be mature enough to provide specifications to create interoperable web services with all the functionality needed to create a distributed application on top of GII. These tools are not only available for the group of experts in our use cases, but they are also remotely accessible by anyone accessing this GII. As we have demonstrated by reusing some of the WPS implemented and deployed both in the AWARE and EuroGEOSS scenarios, finding the right design and implementation criteria (e.g. appropriate granularity and standard interfaces) increases reusability and the potential availability of these processing resources in distributed environments or systems, using standard interfaces and following the technical architecture and INSPIRE guidelines.

INSPIRE defines a technical architecture that seems to guarantee the interoperability between systems and system components. We have been able to reach almost all the user requirements by creating services compliant with INSPIRE service types. However, since INSPIRE's Transformation Type only considers coordinate transformation; we suggest that this service type (geoprocessing) should consider a wider range of functionality.

The scientific processes we have wrapped to expose as web processing services offer functionality not considered by this INSPIRE Service Type. Thus, we propose to offer all kinds of functionality, possibly by the use of standard service profiles, as has been proposed for generalization functionality in other works (Neun et al, 2008).

As the library of reusable web processing services has been implemented within a GMES project and following the INSPIRE technical architecture, this library can have an added value by being reused in broader systems such as GEOSS which aim is to create and connect tools to monitor the environment.

Limitations and lessons learned

To address some of the problems in implementing distributed geospatial processing services, we should mention the overall service chain performance (which is rather variable) when distributed data sources are involved. This is the case when large processing tasks are performed over the network, because data transportation and validation can dramatically increase the response time to users. Friis-Christiensen and colleagues (2007) propose the use of asynchronous messaging to address time-consuming requests. In asynchronous messaging, the WPS instance does not immediately return the process results but rather it responds some time later in a different communication session. These tests will form part of our future work.

Syntactic interoperability in distributed processing is improved by using the OGC WPS specification, but as WPS supports all kind of operations without any requirement in its granularity and does not yet deal with profiles, there is a lack in semantic interoperability(Brauner et al., 2009; Foerster et al., 2010). Not considering WPS profiles makes the processes very general which can bring problems with interoperability at semantic and structural levels. Current work being done within the OGC WG-WPS 2.0 seems to indicate that the new version of this specification will be split into a core specification plus extensions dealing with different profiles.

Furthermore, we would like to mention some other lessons learned along this dissertation: wrapping legacy software has not been a trivial task. In the AWARE use case we faced scientists who own out-of-date software (e.g. old FORTRAN libraries). The process to apply our methodology to wrap this functionality had to perform several steps like wrapping the FORTRAN code in C language in order to be accessible by the Java code using JNI[xxxv] within a 52North WPS. This raises some technical issues about the difficulty in migrating from one platform to another.

Other considerations to take into account include the implementation and deployment of the generated WPS. Regarding the proposed methodology to wrap existing software, the concept is general and the proposed solution tries to address it technologically independently, but the implementation is attached to the current technology. The methodology is based on wrapping existing FOSS under certain conditions, but the current implementation is attached to the WPS implementation provided by 52N. The draft of OGC WPS 2.0 considers the transactional interface that permits the deployment of new processes during execution time in existing WPS instances; this offers more possibilities to assist users in deploying more processes independently of the technology underneath. Work on this is currently being developed by 52N and Erdas[xxxvi].

6.2 Data resource availability

Regarding the improvement of data resources availability, we have presented a framework that follows a hybrid GII building methodology. We provide an INSPIRE-based architecture, whose components assist users in deploying data and information resources as INSPIRE-compliant services participating in the

GII building and maintenance. As a result, users can massively deploy resources in the GII, thus improving availability by maximizing the number of interoperable resources by increasing standardization.

This method permits a rapid resource delivery. The important feature is that resources are delivered in a standard way. As we have seen, the resource can be a local file or a process result. The most innovative fact is that we provide a rapid mechanism to deliver processing results that are deployed easily in standard data visualization and download services. Therefore, the ability to deploy automatically data resources permits to keep GII up-to-date, improving GII usefulness for spatial users' requirements. These two factors, rapid delivery and up-to-date resources, have been pointed out to be key issues in environmental and disaster management scenarios.

There are still many issues that remain open to be addressed in the future:

- Targeted users

The range of targeted users of this approach is quite wide. For instance, government organizations, where, in general only a few employees, or externally-hired personnel, have the technological knowledge to deploy and publish resource. This causes a bottleneck for publishing the new data and generated information to build and maintain GII. Non-government and general users, owning a small quantity of data and want to publish it, with no knowledge or infrastructure in which to deploy and maintain these services.

The proposed mechanism can cope with both types of users. Automatic mechanisms to generate services can assist users with no technical knowledge and can assist, to avoid manual errors, users with more technological knowledge.

- Cloud

Our approach contains a framework to support the deployment and maintenance of services; this is very adequate for general users without their own infrastructure. For government users that have an official infrastructure, the proposed ServiceFramework could be embedded in the existing GII. The presented prototype is tight to a specific technology as we have seen in the implementation section and will not be possible to associate with any service technology. Currently, the proof of concept deploys resources within the geospatial services existing in the ServiceFramework, but this could be changed to deploy the resources in other user geospatial services.

- Security

Automatic deployment mechanisms open a way for general users to publish data and information leading to potential security and privacy issues. In the case of government users, the mechanisms to guarantee data privacy should be the same ones that they use with a manual mechanism. The ServiceFramework deals with improving the availability of data by providing automatic mechanisms, but it does not deal with security issues at this scale. Security issues are not yet taken into account and it is something to consider in the future like validating users and their rights.

Limitations and lessons learned

- OWS-T: Transactional service interfaces.

To deploy resources automatically as standard services, the ServiceFramework uses a concrete implementation of the OGC WMS, WFS, WCS specification. The ServiceFramework uses Geoserver's own protocol to deploy resources. We still could migrate the solution to other implementations, but some programming should be done. To avoid this, it would be useful to deploy resources using standard transactional operations. This would require to have the OGC WMS and WCS specifications including a transactional interface like WFS and WPS is considering in its currently and future versions. This would detach the solution from the vendor implementation.

- Technology dependency and interoperability

Although using the same standards, we still find syntactic interoperability problems regarding the different versions of each standard implemented by each vendor. For instance, if the user wants to integrate new vector data, raw or processed (Shapefile, GML), and the ServiceFramework deploys it as OGC WFS by using the Geoserver implementation, by default this implementation is OGC WFS 1.1.0 which returns GML 3. Later when we retrieve this FeatureCollection from Geoserver to be used as input to one of the OGC WPS implemented by using the 52North framework, it fails because this implementation only accepts GML inputs in GML 2.x. Thus we have to specify that we request OGC WFS version 1.0.0 so that we retrieve the interoperable GML with the implementation of our WPS.

GML interoperability has been one of the biggest challenges to overcome, since it is a complex format that each vendor implements in their own way (Lu et al, 2007).

- Semantic interoperability

GII provides the infrastructure in which spatial wrappers and mediators play a facilitating role. WMS and WFS services wrap the data sources, abstracting data from its machine representation, and become accessible to diverse users in a uniform way. The adoption of these OGC interfaces and standards makes possible spatial data integration in a distributed environment when semantic differences are not too great. Similarly as mentioned in section 6.1 where we describe that the used OGC WPS specification offers the abstraction at syntactic level to offer interoperable operations in the GII. The further work with WPS profiles would provide some semantics on the processes offered.

Therefore, future research on semantic interoperability is needed to reach generally acceptable levels of *ad hoc* spatial data and process integration. Relevant to this line we would like to mention the work of Lemmens (2006) regarding semantic interoperability of distributed geospatial services and Janowicz et al (2010) about semantics en SDI. Other promising works relevant for addressing semantics in GII are collected in (van Oosterom and Zlatanova, 2008).

- Data formats

The current proof of concept of the ServiceFramework is able to assist users in deploying data with spatial formats that are either supported natively by the chosen implementation of the standard data services or transformed previously to supported formats by the ServiceFramework. For instance Geoserver does not currently support GML data sources, so to deploy GML datasets, the ServiceFramework transforms it to Shapefile before deploying it.

Further research is needed to find a common format to encapsulate (adding georeferencing) to any resource provided by user. For instance, current research is being performed to wrap resources using KML. This would help to be able to deploy any resource as an INSPIRE service.

- Symbology

Simbology deployment is not yet considered. The ServiceFramework user interface allows users to choose a solid colour for visualization. Further research and implementation should be done in order to generate a set of automatic styles so users could easily choose one to be deployed in the OGC WMS, for instance generating an OGC SLD[xxxvii].

- Metadata

As we see in the Service Publisher section, only a minimum set of metadata elements for service description are being generated and published in a Service Catalogue in our proposal. As we have mentioned, there are software packages able to automatically generate some standard metadata elements from a spatial data source (Díaz et al, 2007). This software could be integrated within the ServiceFramework in the future. Another way to generate metadata to publish the data deployed in the OGC Data Services is querying the Services themselves by using the OGC interfaces that provide information about the data elements they serve.

- Quality

In regard to general users and the opening of GII to share and publish geospatial data, unresolved issues of quality are raised. We would like to emphasize the need to increase the availability of data and information in an interoperable way because nowadays there is a lack of these resources. So we point out the need to have at least poor quality data as a better option of not having data. Later on, when the issue of data availability is solved, we can approach the problem of measuring the quality of data provided by mass-market users by methods like data ratting, etc.

Further research is being done, within the EuroGEOSS project (Leibovici et al, 2009), to manage metadata elements to add some information about the quality of the resources.

- Open Source

The use of Open source components has the advantage of giving us full control over the code to modify everything. A disadvantage, however, is that they have caused some inconveniences because most of them use libraries that are not fully developed in the context of this project. Therefore, we had to be part of the developer team to solve bugs.

Also, implementations of the standard specifications do not provide useful error messages regarding the types of problems. Documentation scarcity is always a big issue to face.

6.3 Resource visibility

Improving visibility of geospatial resources means enhancing the capability of the resources to be found. Metadata are necessary to describe resources, and together with Catalogue Services, they are the key elements for discovery and resource fusion possibilities (Nogueras et al., 2005; Díaz et al., 2007). INSPIRE directive mandates the creation and maintenance of metadata and related Discovery Services (Craglia et al., 2007). For this purpose we describe a mechanism to generate and publish the metadata of resources in Catalogue Services in GII in a semi-automatic way. The Resource Publisher module provides this capability.

As a future work, as it is an ongoing work with the metadata subgroup of the Spanish SDI. Another approach is to generate automatically metadata of geospatial resources once they have been deployed as standard services. We can extract information from the OGC documents like *GetCapabilities* but also further information about the Service content (e.g. *DescribeFeatureType*, *DescribeLayer* and *DescribeProcess* documents) to generate other standard-formatted metadata (e.g. ISO and its profiles like INSPIRE).

Limitations and lessons learned

To publish data resources in Catalogue Services is out of the scope of this work and we have previous works with open source tools that could be integrated as part of the Resource Publisher.

We focus on the metadata created for processes and services, using the Capabilities documents for publishing processes where the ResourcePublisher publishes them transparently in Catalogue Services.

For the first prototype we have chosen Geonetwork, as implementation of Catalogue Service and the Resource Publisher publishes the metadata using Geonetwork protocol. In the present, we are extending the prototype to use other OGC CSW (Nebert and Whiteside, 2004) transactional implementations.

For future work, we consider the automatic generation of more sophisticated metadata by querying the deployed resources, creating metadata that contains the format, date, online resource, etc., as well as publishing them using the CS-W transactional interface where users could choose any target Catalogue Service.

Bibliography

Aalst, W.M.P. van der, Beisiegel, M., Hee, K.M. van der, Konig, D., Christian Stahl, C., 2007. An SOA-based architecture framework. International Journal of Business Process Integration and Management 2 (2) 91-101.

Abel, D.J. 1998. Towards integrated geographical information processing. int. j. geographical information science 12 (4) 353-371

Anderson, G., Moreno-Sanchez, R. 2003. Building Web-Based Spatial Information Solutions around Open Specification and Open Source Software. Transaction in GIS 7 (4) 447-446

Abdalla, R. , Tao, V., Li, J., 2007. Challenges for the application of GIS Interoperability in Emergency Management. In: Lecture Notes in Geoinformation and Cartography. Editors Jonathan Li, Sisi Zlatanova, Andrea Fabbri City: New York Publisher: Springer-Verlag: 389–405

Alameh, N. 2003. Chaining Geographic Information Web Services. IEEE Internet Computing 7(5):22-29

Almer, A., Schnabel, T., Granica, K., Hirschmugl, M., Raggam, J., van Dahl, M. 2008. Information Services to Support Disaster and Risk Management in Alpine Areas. The European Information Society: 415-432.

Alonso, G., Casati, F., Harumi, K., Machiraju, V., 2004. Web Services. Concepts, Architectures and Applications. Heidelberg: Springer.

Annoni, A. 2007. INSPIRE/GMES/GEOSS. Head of Spatial Data Infrastructures Unit. Technical report. Orchestra Day. 12, December 2007.

Arnell, N., Chunzhen, L. 2001. Hydrology and Water Resources. In IPCC third assessment report – climate change: impacts, adaptations, and vulnerability. CUP, Cambridge 191-234.

Bayarri, S., Capo, Y. 2010. Lessons, impact and future prospects after the implementation of a disaster-risk management SDI in the Andean Community. Proceedings of Gi4DM 2010 Conference on Geomatics for Crisis Management (Gi4DM 2010). Torino, Italy, Feb 2010.

De La Beaujardiere, J. 2004. Web Mapping Service Implementation Specification, Version 1.3.0. Open Geospatial Consortium. Document available at http://www.opengeospatial.org/standards/wms

Béjar, R., Latre, M. Á., Nogueras-Iso, J., Muro-Medrano, P. R., Zarazaga-Soria, F. J. 2009a. Systems of Systems as a Conceptual Framework for Spatial Data Infrastructures. International Journal of Spatial Data Infrastructures Research 4 201-217

Béjar, R., Latre, M. Á., Nogueras-Iso, J., Muro-Medrano, P. R., Zarazaga-Soria, F. J. 2009b. An architectural style for Spatial Data Infrastructures. International Journal of Geographical Information Science, 23(3), 271-294.

Bernard, L, Einspanier, U., Lutz, M., Portele, C. 2003. Interoperability in GI Service Chains The Way Forward. In: M. Gould, R. Laurini & S. Coulondre (Eds.). 6th AGILE Conference on Geographic Information Science 2003, Lyon: 179-188

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

Bernard, L., Kanellopoulos, I., Annoni, A., Smits, P., 2005. The European geoportal--one step towards the establishment of a European Spatial Data Infrastructure. Computers, Environment and Urban Systems 29 (1) 15-31.

Bernard, L., Craglia, M., Gould, M., Kuhn, W. 2005. Towards an SDI Research Agenda. 11 EC-GIS (Sardinia), June 28- July 1, 2005.

Berners-Lee, T., Hall, W., Hendler, J., Shadbolt, N., Weitzner, D.J., 2006. Creating a Science of the Web. Science 313 769-771.

Best, B. D., Halpin, P. N., Fujioka, E., Read, A.J., Qian, S.S., Hazen, L.J., Schick, R.S., 2007. Geospatial web services within a scientific workflow: Predicting marine mammal habitats in a dynamic environment. Ecological Informatics 2 (3) 210-223.

Bishop, I.D., Escobar, F. J., Karuppannan, S., Suwarnarat, K., Williamson, I.P., Yates, P.M., Yaqub, H.W. 2000 Spatial data infrastructures for cities in developing countries: Lessons from the Bangkok experience. Cities 17(2) 85-96.

Bishr, Y. 1998. Overcoming the semantic and other barrier to GIS interoperability. International Journal of Geographical Information Science 12 (4) 299-313.

Booth, D., et al. (eds). 2003 Web Services Architecture (W3C,Working Draft, 2003; www.w3.org/TR/2003/WD-wsarch- 20030808).

Budhathoki, N. R., Bertram, B., et al. 2008. Reconceptualizing the role of the user of spatial data infrastructure. GeoJourna 72 149-160.

Budhathoki, N. R., Nedovic-Budic, Z. 2007. Expanding the SDI knowledge base. In H. Onsrud (Ed.), Research and theory in advancing spatial data infrastructure.Redlands: ESRI Press. 7–32

Buehler, K., McKee, L., 1996. The OpenGIS Guide. Introduction to Interoperable Geoprocessing. Part I of the Open Geodata Interoperability Specification (OGIS). OGIS TC Document 96-001, OGIS Project 6 Technical Committee of the OpenGIS Consortium Inc

Bulterman D. 2004. Is it Time for a Moratorium on Metadata? IEEE Multimedia, October-December (2004) 10-17.

Brauner, J., Schäffer, B. 2008. Integration of GRASS functionality in Web Based SDI service chains. Academic Proceedings of the 2008 Free and Open Source Software for Geospatial Conference, Cape Town, South Africa.

Brauner, J., Foerster, T., Schaeffer, B., Baranski, B. 2009. Towards a Research Agenda for Geoprocessing Services. AGILE Proceedings, Hannover, Germany.

Brunner, D., Lemoine, G., Thoorens, F., Bruzzone, L. 2009. Distributed Geospatial Data Processing Functionality to Support Collaborative and Rapid Emergency Response. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 2 (1) 33-46

Caldeweyher, D., Zhang, J., Pham, B. 2006. OpenCIS–Open Source GIS-based web community information system. International Journal of Geographical Information Science 20 (8) 885-898

Canfora, G., Fasolino, A.R., Frattolillo, G., Tramontana, P. 2008. A wrapping approach for migrating legacy system interactive functionalities to Service Oriented Architectures. The Journal of Systems and Software, 81, 463–480

Carrera, F., Ferreira, J. 2007. The future of spatial data infrastructures: Capacity-building for the emergence of municipal SDIs. International Journal of Spatial Data Infrastructures Research 2 54–73.

Cepický, J., Becchi, L. 2007 Geospatial Processing via Internet on Remote Servers - PyWPS. OSGeo Journal, vol 1. Available at http://www.osgeo.org/journal/volume1

Chang-Tien Lu, Raimundo F. Dos Santos, Lakshmi N. Sripada, Yufeng Kou. 2009. Advances in GML for Geospatial Applications. GeoInformatica 11 (1) 131-157

Cheng, Q., Ko, C., Yuan, Y., Ge, Y., Zhang, S. 2006. GIS modeling for predicting river runoff volume in ungauged drainages in the Greater Toronto Area, Canada. Computers & Geosciences 32 (8) 1108-1119

Chew, L. P. Voronoi/Delaunay Applet. http://www.cs.cornell.edu/Info/People/chew/chew.html (Accessed July 2008)

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

Chow, T.E., 2008. The Potential of Maps APIs for Internet GIS Applications. Transactions in GIS, 12 (2) 179-191.

Cox, S., Cuthbert, A., Lake, R., Martell, R. Eds. 2002. OpenGIS® Geography Markup Language (GML) Implementation Specification, version 2.1.2. OpenGIS® Implementation Specification, 17 September 2002. OpenGIS Project Document Number 02-069

Craglia, M., Annoni, A., Masser, I. (Eds.), 1999. Geographic Information Policies in Europe: National and Regional Perspectives. EUROGI-EC Data PolicyWorkshop , Amersfoort, 15 November 1999. European Commission—Space Applications Institute, European Communities, available at http://www.ec-gis.org/reports/policies.pdf.

Craglia, M. 2007. Volunteered Geographic Information and Spatial Data Infrastructures: when docparallel lines converge? Specialist Meeting on Volunteered Geographic Information. 13-14 December 2007. Santa Barbara. Retrieved January, 2010, from http://www.ncgia. ucsb.edu/projects/vgi/participants.html

Craglia, M., Kanellopoulos, I., Smits, P. 2007. Metadata: where we are now, and where we should be going. Proceedings of 10th AGILE International Conference on Geographic Information Science 2007. Aalborg University, Denmark

Craglia, M., Goodchild, M.F., Annoni, A., Câmara, G., Gould, M., Kuhn, W., Mark, D.M., Masser, I., Maguire, D.J., Liang, S., Parsons, E. 2008. Next-generation Digital Earth. A position paper from the Vespucci Initiative for the Advancement of Geographic Information Science. International Journal of Spatial Data Infrastructure Research 3 146–167.

Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., Weerawarana, S., 2002. Unravelling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI. IEEE Internet Computing 6 (2) 86–93.

Davis, C.A. Jr., Fonseca, F.T., Câmara, G. 2009. Beyond SDI: Integrating Science and Communities to Create Environmental Policies for the Sustainability of the Amazon. International Journal of Spatial Data Infrastructures Research, 4, 156-174

de Groot, W., Goldammer, J.G., Keenan, T., Brady, M.A., Lynham, T.J., Justice, C.O., Csiszar, I.A., O'Loughlin, K. 2006. Developing a global early warning system for wildland fire. Forest Ecology and Management, 234 (1) 10

de Sarriá-Sopeña, S., Yebra-Valverde, R.T. & Mendoza-Domínguez, P., 2007. Sistema Integrado para la Gestión y Dirección de Incendios Forestales en Andalucía (SIGDIF). En Sevilla, Spain. Available at: http://www.fire.uni-freiburg.de/sevilla-

2007/contributions/doc/SESIONES_TEMATICAS/ST7/deSarria_et_Al_SPAIN_Andal_SIGDIF.pdf.

Denzer, R., 2005. Generic integration of environment decision support systems – state-of-the-art. Environmental Modelling & Software 20 (10) 1217-1223.

Di, L., Zhao, P., Han, W., Wei, Y., Li, X. 2007. GeoBrain Web Service-based Online Analysis System (GeOnAS). In Proc. of NASA Earth Science Technology Conference.

Di Lorenzo, G., Fasolino, A.R., Melcarne, L., Tramontana, P., Vittorini, V. 2007. Turning Web Applications into Web Services by Wrapping Techniques. Univ. di Napoli Federico II, Naples. 14th Working Conference on Reverse Engineering, 2007. WCRE 2007.

Dienel, P., 1989. Contributing to social decision methodology: citizen reports on technological projects. In: Vlek, C., Cvetovich, G. (Eds.), Social Decision Methodology for Technological Projects (Theory and Decision Library, Series A). Kluwer Academic Publishers.

Díaz L, Costa S, Granell C, Gould M. 2007a. Migrating geoprocessing routines to web services for water resource management applications. In Proceedings of 10th AGILE Conference on Geographic Information Science (AGILE 2007), Aalborg (Denmark)

Díaz, L., Martín, C., Gould, M., Granell, C., Manso, M.A. 2007b. Semi-automatic Metadata Extraction from Imagery and Cartographic data, International Geoscience and Remote Sensing Symposium (IGARSS 2007). Barcelona, Julio 2007. IEEE CS Press, pp. 3051-3052.

Díaz, L., Gould, M., Beltrán, A., Llaves, A., Granell, C. 2008a. Multipurpose metadata management in gvSIG. In Academic Proceedings of the 2008 Free and Open Source Software for Geospatial Conference (FOSS4G 2008). Cape Town, South Africa, Sep 2008, pp. 90-99

Díaz, L., Granell, C., Gould, M., Olaya, V. 2008b. An open service network for geospatial data processing. In Academic Proceedings of the 2008 Free and Open Source Software for Geospatial Conference (FOSS4G 2008). Cape Town, South Africa, Sep 2008, pp. 410-419, ISBN 978-0-620-42117-1

Díaz, L., Granell, C., Gould, M. 2008c. Case Study: Geospatial processing services for web-based hydrological applications. In J.T. Sample, K. Shaw, S. Tu, M. Abdelguerfi (Eds.): Geospatial Services and Applications for the Internet. Springer. ISBN 978-0-387-74673-9.

Díaz, L., Granell, C., Gould, M. 2009a. Spatial Data Integration over the Web. In V. E. Ferraggine, J.H. Doorn, L.C. Rivero (Eds.) Handbook of Research on Innovations in Database Technologies and Applications: Current and Future Trends, Information Science Reference (Hershey, . 2009), 2 325-333.

Díaz, L., Granell, C., Huerta, J., Simonazzi, W. 2009b. Global initiatives for Earth Observation: European contributions to GEOSS. Upgrade: The European Journal for the Informatics Professional, X(2) 29-36

Díaz, L., Aragó, P., Granell, P., Huerta, J. 2010a. Integrating user-generated information for emergency management. Proceedings of Gi4DM 2010 Conference on Geomatics for Crisis Management (Gi4DM 2010). Torino, Italy, Feb 2010.

Díaz, L., Granell, C., Gould, M., Huerta, J. 2010b. User-driven generation of standard data services. Accepted to European Geosciences Union (EGU) General Assembly 2010 (EGU 2010). Viena, Austria, May 2010.

Díaz, L., Granell, C., Gould. 2010c. Enhancing the visibility and availability of geospatial resources in disaster management information systems. Submitted to International Journal of Applied Earth Observation and Geoinformation (under review)

Díaz, L., Granell, C., Gil, J. Huerta, J. 2010d. Managing user generated information in geospatial cyberinfrastructures. Submitted to Future Generation Computer Systems. The International Journal of Grid Computing and eScience (under review)

Diehl, S., Neuvel, J., Zlatanova, S. and Scholten, H. 2006, Investigation of user requirements in the emergency response sector: the Dutch case, in: Second Symposium on Gi4DM, 25-26 September, Goa, India

Dzemydiene, D., Maskeliunas, S., Jacobsen, K. 2008 Sustainable management of water resources based on web services and distributed data warehouses. Baltic Journal on Sustainability. 14 (1) 38:50

European Commission (IDABC), European Interoperability Framework for Pan-European e-Government Services (2004). Available at: http://ec.europa.eu/idabc/en/document/3761 (accessed 21 February 2010).

EuroGEOSS WP3. 2009. D.3.1: Report on user requirements for the

EuroGEOSS Forestry operating capacity. Available at http://www.eurogeoss.eu/wp/wp1/Deliverables/EuroGEOSS_D3-1.pdf

EuroGEOSS WP3. 2010. D.3.2: Design Specifications for EuroGEOSS Forestry Components and Interfaces. Available at http://www.eurogeoss.eu/wp/wp1/Deliverables/EuroGEOSS_D3-2.pdf

EuroGEOSS WP2. 2010. D.2.2.2. Specification of EUROGEOSS Initial Operating Capacity. Available at: http://www.eurogeoss.eu/wp/wp1/Deliverables/EuroGEOSS_D_2_2_2.pdf

EuroGEOSS, 2010, “The EuroGEOSS Broker, technical documents”, available at http://www.eurogeoss.eu/

Feng, P. 2003. Studying standardization: a review of the literature. In Procedings of the 3rd conference on Standarization and Innovation in Information Technology (SIIT2003), IEEE 99-112

Feuerlicht, G. 2006. Service granularity considerations based on data properties of interface parameters. Computer Systems Science and Engineering 21 (4) 315-320.

Fileto, R. 2001. Issues on Interoperability and Integration of Heterogeneous Geographical Data. . Proceedings of 3rd Brazilian Symposium on GeoInformatics.

Foster, I., Kesselman, C., Tuecke, S., 2001. The anatomy of the Grid: Enabling scalable virtual organisations. International Journal of Supercomputer Applications 15 200-222.

Foster, I., Alpert, E., Chervenak, A., Drach, B., Kasselman, C., Nefedova, V., Middleton, D., Shoshani, A., Sim, A., Williams, D. 2002. The Earth System Grid II: Turning climate datasets into community resources, 19th Conference on Interactive Information Processing Systems, American Meteorological Society, Long Beach, CA,.

Foster, I. 2005. Service-Oriented Science. Science 308: 814-017

Foerster, T. 2006 An open software framework for web service-based geo-processing. In Proceedings of the Free and Open Source Software for Geospatial (FOSS4G 2006), Lausanne (Switzerland).

Foerster, T., Shäffer, B. 2007. A Client for Distributed Geo-processing on the Web. In Proceedings of International Symposium on Web and Wireless Geographical Information Systems (W2GIS 2007), Cardiff (UK). Berlin: Springer LNCS 4857, pp. 252-263.

Foerster, T., Lehto, L., Sarjakoski, T., Sarjakoski, L.T., Stoter, J. 2010. Map generalization and schema transformation of geospatial data combined in a Web Service context. Computers, Environment and Urban Systems 34 79–88

Fook, K.D., Vieira-Monteiro, A.M., Câmara, G., Casanova, M.A., Amaral, S. 2009 Geoweb Services for Sharing Modelling Results in Biodiversity Networks. Transactions in GIS, 13 379-399.

Freeman, E., Sierra, K., Bates, B. Head First Design Patterns, O'Reilly, 2004.

Friis-Christensen, A., Bernard, L., Kanellopoulos, I., Nogueras-Iso, J., Peedell, S., Schade, S., Thorne, C. 2006. Building Service Oriented Applications on top of Spatial Data Infrastructure- A Forest Fire Assessment Example. Agile, Visegrad, Hungary.

Friis-Christensen, A., Lutz, M., Ostländer, N., Bernard, L. 2007. Designing Service Architectures for Distributed Geoprocessing:challenges and future directions. Transactions in GIS 11(6) 799-818

Friis-Christensen, A., Lucchi, R., Lutz, M, Ostländer, N. 2009. Service chaining architectures for applications implementing distributed geographic information processing. International Journal of Geographical Information Science, 23 (5) 561-580.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. 1995. Design Patterns. Addison-Wesley

Garcia-Molina H, Hammer J, Ireland K, Papakonstantinou Y, Ullman J, Widom J. 1995. Integrating and accessing heterogeneous information sources in TSIMMIS. Proceedings of the AAAI Symposium on Information Gathering, Stanford, California: 61-64

GEOSS Secretariat. 2007. The first 100 steps to GEOSS. Ed: GEOSS Secretariat. 30 November 2007

GEOSS. 2008. WHITE PAPER ON THE GEOSS DATA SHARING PRINCIPLES. Available at: http://www.earthobservations.org/documents/dsp/Draft%20White%20Paper%20for%20GEOSS%20Data%20Sharing%20Policies_27Sept08.pdf

Giovando, C., Whitmore, C., Camia, A., San Miguel, J., Boca, R., Lucera, J. 2010. GEOSPATIAL SUPPORT TO FOREST FIRE CRISIS MANAGEMENT AT THE EUROPEAN LEVEL. Proceedings of Gi4DM 2010 Conference on Geomatics for Crisis Management. Torino, Italy, Feb 2010.

Gonzalez-Escalante, L., Botello-Castillo, A., Balladares-Ocana L. 2005. Automatic Generation and Publication of Web Services for the Access and Integration of Distributed Data Sources. Sixth Mexican International Conference on Computer Science (ENC'05), 96-105

Goodall, J.L., Horsburgh, J.S., Whiteaker, T.L., Maidment, D.R., Zaslavsky, I., 2008. A first approach to web services for the National Water Information System. Environmental Modelling & Software 23 (3) 304-401.

Gould, M. 2007. Vertically interoperable geo-infrastructures and scalability. Specialist Meeting on Volunteered Geographic Information. 13-14 December 2007. Santa Barbara. Retrieved January, 2010, from http://www.ncgia.ucsb.edu/projects/vgi/participants.html

Gould, M. 2006a. Meta - Findability: Part 2. GeoConnexion, September 2006, 28-29.

Gould, M. 2006b Meta - Findability: Part 1. GeoConnexion, July/August 2006, pp. 36-37.

Gould, M., Granell, C., Esbrí, M.A, Díaz, L., Manso, M.A. 2006. Simplifying Geospatial Metadata Collection and Exploitation. In Abstract Handbook of the 9th International Conference for Global Spatial Data Infrastructure (GSDI-9). Santiago de Chile (Chile), November 2006.

Goodchild, M.F., M.J. Egenhofer, R. Fegeas, and C.A. Kottman (eds). 1999. Interoperating Geographic Information Systems. Boston: Kluwer Academic Publishers.

Goodchild, M.F. 2008. Geographic information science: the grand challenges. In J.P. Wilson and A.S. Fotheringham, editors, The Handbook of Geographic Information Science. Malden, MA: Blackwell, pp. 596–608.

Goodchild, M.F. 2007. Citizens as voluntary sensors: spatial data infrastructure in the world of Web 2.0. International Journal of Spatial Data Infrastructures Research 2 24–32.

Gore A., 1999. The Digital Earth: Understanding our planet in the 21st Century. Photogrammetric Engineering and Remote Sensing 65 (5) 528

Granell, C., Gould, M., Gronmo, R., and Skogan, D. 2005. Improving reuse of web service compositions, in Proceedings of the International Conference of E-Commerce and Web Technologies, pp. 358-367.

Granell, C., Díaz, L., Gould. M. 2007. Managing Earth Observation data with distributed geoprocessing services. In Proceedings of the International Geo-science and Remote Sensing Symposium (IGARSS 2007). Barcelona (Spain), July 2007.

Granell, C., Díaz, L., Gould, M. 2008. Distributed Geospatial Processing Services. In M. Khosrow-Pour (Eds.) Encyclopedia of Information Science and Technology, Second Edition, Information Science Reference (Hershey, 2008), pp. 1186-1193, ISBN 978-1-60566-026-

Granell, C., Díaz, L., Gould, M. 2008. Geoservice Implementation Report. Public Deliverable D252.2 available at the AWARE Web site. www.aware-eu.info

Granell, C., Díaz, L., Gould, M. 2010. Service-oriented applications for environmental models: Reusable geospatial services. Environmental Modelling and Software, 25(2): 182-198, 2010. ISSN: 1364-8152

Groot, R., McLaughlin, J. 2000. Geospatial Data Infrastructure, Concepts, Cases and Good Practice, Oxford University Press.

Group on Earth Observations. 2005. The GEOSS 10 Year Implementation Plan. GEO document 1000, February 2005. http://www.earthobservations.org/

GEOSS. Global Earth Observation System of Systems http://www.earthobservations.org/

GMES. Global Monitoring for Environment and Security. http://www.gmes.info/

Haesen, R., Snoeck, M., Lemahie, W., Poelmans, S., 2008. On the Definition of Service Granularity and Its Architectural Impacts. In Proc. of International Conference on Advanced Information Systems Engineering (CAiSE'08). Springer, LNCS 5078, 375-389.

Hey, T., Trefethen, A. E., 2005. Cyberinfrastructure for e-Science. Science 308 817-821.

Hood, C. A. 2005. Implementation of GEOSS: A review of all-hazards warning system and its benefits to public health, energy, and the environment, Written testimony, U.S. House Committee on Energy and Commerce, 9 March 2005.

Information Systems and Disaster Risk Reduction. World Summit on the Information Society. Ginebra 2003 – Túnez 2005. www.unisdr.org/news/WSIS/WSIS.pdf

Iannotta, B. 2007. GEOSS : A global view. Aerospace America. August 2007. pag 36-40.

INSPIRE EU Directive. 2007. Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE). Official Journal of the European Union, L 108/1, Volume 50, 25 April 2007. http://eur-lex.europa.eu/JOHtml.do?uri=OJ:L:2007:108:SOM:EN:HTML.

INSPIRE Network Services Drafting Team. 2007. INSPIRE Network Services Architecture

INSPIRE Network Services Drafting Team. 2008. Draft Technical Guidance to implement INSPIRE View Services

INSPIRE Network Services Drafting Team. 2009.Draft Technical Guidance for INSPIRE Download Services

INSPIRE Network Services Drafting Team. 2009b. Technical Guidance for INSPIRE Discovery Services.

ISO, 2003 ISO, 2003. Geographic Information—Metadata. ISO 19115:2003, International Organization for Standardization (ISO).

Jacoby, S.K., Smith, J.C., Ting, L.A., Williamson, I.P. 2002. Developing a common spatial data infrastructure between State and Local Government - an Australian case study. International Journal of Geographical Information Science, Taylor & Francis Ltd(London), 16 (4) 305-322

Jankowski, P. 2009. Towards participatory geographic information systems for community-based environmental decision making. Journal of Environmental Management 90. 1966–1971

Janowicz, K., Schade, S., Bröring, A., Keßler, C., Maue, P. and Stasch, C. 2010. Semantic Enablement for Spatial Data Infrastructures. Accepted as article in Transactions in GIS 14(2).

Jeong, S., Liang, Y., Liang, X., 2006. Design of an integrated data retrieval, analysis, and visualization system: Application in the hydrology domain. Environmental Modelling & Software 21 (12) 1722-1740.

Kiehle, C. 2006. Business logic for geoprocessing of distributed geodata. Computers & Geosciences, 32(10): 1746–1757.

Kiehle, C., Greve, K., and Heier, C. 2006. Standardized Geoprocessing - Taking Spatial Data Infrastructures one Step Further. 9th AGILE Conference on Geographic Information Science, Visegrád, Hungary, 2006

Kiehle, C.; Heier, C. & Greve, K. 2007. Requirements for Next Generation Spatial Data Infrastructures-Standardized Web Based Geoporcessing and Web Service Orchestration. Transactions in GIS, Blackwell Publishing, 2007, 11, 819-834

Ko, I.-Y., Neches, R., 2003. Composing Web Services for Large-Scale Tasks. IEEE Internet Computing 7 (5) 52-59.

Kralidis.A.T. 2007. Geospatial Web Services: The Evolution of Geospatial Data Infrastructure. Scharl, K. Tochtermann (Eds.) THE GEOSPATIAL WEB How Geobrowsers, Social Software and the Web 2.0 are Shaping the Network Society. Springer:London.

Krishnan, S., Stearn, B., Bhatia, K., Baldridge, K.K., Li, W., Arzberger, P. 2006. Opal: Simple Web Services Wrappers for Scientific Applications. In proceedings of ICWS 2006, IEEE International Conference on Web Services

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

Larman, C., Basili, V.R. 2003. Iterative and Incremental Development: A Brief History. IEEE Computer (IEEE Computer Society) 36 (6) 47–56.

Lawrence, B. 2003. Experiences with archiving databases in British Atmospheric Data Center, 6th DPC Forum focussed on Open Source Software and Dynamic Databases, London, U.K., 24 June 2003.

Lee, C., Percivall, G. 2008. Standards-Based Computing Capabilities for Distributed Geospatial Applications. Computer 41, 11 (Nov. 2008)

Lacerda-Alves, L., Davis, C.A. Jr. 2006. Evaluation of OGC Web Services for Local Spatial Data Infrastructures and for the Development of Clients for Geographic Information Systems. Advances in Geoinformatics VIII Brazilian Symposium on GeoInformatics, GEOINFO 2006, Campos do Jordão (SP), Brazil, November 19–22, 2006. ISBN: 978-3-540-73413-0 (Print) 978-3-540-73414-7 (Online) Part 4. Pag 217-234. Springer Berlin Heidelberg 2007

Leibovici, D., Hobona, G., Stock, K. and Jackson, M. 2009. Qualifying geospatial workfow models for adaptive controlled validity and accuracy. In: IEEE proceedings 17th International conferenceon GeoInformatics, August 2009, USA, pp. 1-5

Lemmens, R.L.G. 2006. Semantic interoperability of distributed geo - services. Delft, Nederlandse Commissie voor Geodesie (NCG), 2006. 0165-1706Netherlands Geodetic Commission NCG : Publications on Geodesy : New Series 63, 290 p. ISBN: 90-6132-298-7

Lemmens R, Wytzisk A, de By R, Granell C, Gould M, van Oosterom P. 2006. Integrating Semantic and Syntactic Description to Chain Geographic Services. IEEE Internet Computing 10 (5) 42-52

Li, M., Yu, B., Qi, M., Antonopoulos, N. 2008. Automatically wrapping legacy software into services: A grid case study. Peer-to-Peer Networking and Applications, 1 (2) 139-147.

Liu, Y., Gupta, H., Springer, E., Wagener, T., 2008. Linking science with environmental decision making: Experiences from an integrated modeling approach to supporting sustainable water resources management. Environmental Modelling & Software 23 (7) 846-858.

Lu, C-T, Dos Santos Jr, R. F., Sripada, L. N., Kou, Y., 2007. Advances in GML for Geospatial Applications. Geoinformatica 11 (1) 131-157.

Lutz, M., 2007. Ontology-based descriptions for semantic discovery and composition of geoprocessing services. GeoInformatica 11 (1) 1-36.

Maguire, D. J., P. A. Longley, 2005. The emergence of geoportals and their role in spatial data infrastructures. Computers, Environment and Urban Systems 29 (1) 3-14.

Maguire, D. (2006). “Geographic Earth Explorers: A New Software Paradigm for Visualizing and Analyzing Geography?” 2006 Annual Meeting of the Association of American Geographers. Chicago, IL.

W.H.E.D. Man. 2006. Understanding SDI; complexity and institutionalization. International Journal of Geographical Information Science, 20 (3) 329–343.

Manso, M.A., Nogueras, J., Zarazaga, J., Bernabé, M.A. 2004. Automatic Metadata Extraction from Geographic Information, in Proceedings 7 AGILE

Conference on Geographic Information Science, University of Crete, Greece. 379-385.

Mansourian, A., Rajabifard, A., Zoej, M.V. 2005. Development of a Web-Based GIS Using SDI for Disaster Management, in: Geo-information for Disaster Management. 599-608.

Masser, I. 2005. GIS Worlds: Creating Spatial Data Infrastructures. Redlands: ESRI Press.

Masser, I., Rajabifard, A., Williamson, I. 2007. Spatially enabling governments through SDI implementation. International Journal of Geographical Information Science. 22 (1) 5–20

Mineter, M.J., Jarvis, C.H., Dowers, S., 2003. From stand-alone programs towards grid-aware services and components: a case study in agricultural modelling with interpolated climate data. Environmental Modelling & Software 18 (4) 379-391.

Mysiak, J., Giupponi, C., Rosato, P. 2005. Towards the develpment of a decision support system for water resource management, Envorinmental Modelling & Software 20(2):203-214

Mykkänen, J. & Tuomainen, M. 2008. An evaluation and selection framework for interoperability standards. Information and Software Technology, 50(3), 176-197.

McIlraith, S. A., Son, T. C., Zeng, H., 2001. Semantic Web Services. IEEE Intelligent Systems 16 (2) 46-53.

Michaelis, C.D., Ames, D.P. 2009. Evaluation and Implementation of the OGC Web Processing Service for Use in Client-Side GIS. GeoInformatica 13 (1) 109-120

Milanovic, N., Malek, M. 2004. Current Solutions for Web Service Composition. IEEE Internet Computing 8 (6) 51-59.

Moreno-Sanchez, R., Anderson, G., Cruz, J., Hayden, M. 2007. The potential for the use of Open Source Software and Open Specifications in creating Web-based cross-border health spatial information systems. International Journal of Geographical Information Science 21 (10) 1135-1163.

National Research Council (NRC), 2007, Successful response starts with a map, Improving geospatial support for disaster management, The National Academic press, Washington DC, USA, 184 p

Nativi, S., Bigagli, L., Mazzetti, P., Cuomo, V. () Applying SOA to Earth Observation: the COS(0T) experience. Proceedings of ICTTA'04, Damascus (Syria), April 2004, IEEE Press, ISBN 0-7803-8482-2/04, 2004

Nativi, S., Mazzetti, P., Guzzetti, F., Oggioni, A., Pirrone, N., Santoleri, R., Tartari, G. Viola, A. 2010. The GIIDA project: a spatial information infrastructure for environmental data sharing. Proceedings of Gi4DM 2010 Conference on Geomatics for Crisis Management (Gi4DM 2010). Torino, Italy, Feb 2010.

Nature. 2006. Mapping disaster zones: Commentary. Nature, 439, 787–788.

Nayak, S., Zlatanova, S. (Eds.). 2008. Remote Sensing and GIS Technologies for Monitoring and Prediction of Disasters. X 272 p, Springer.

Nebert, D., Whiteside, A. 2004. OpenGIS—catalogue services specification (version: 2.0). OpenGIS Project Document 04-021r2, Open GIS Consortium.

Nebert, D. Ed. 2004. GSDI. Developing Spatial Data Infrastructures: The SDI Cookbook.

Neun, M.; Burghardt, D. and Weibel, R. 2008. Web Service Approaches for Providing Enriched Data Structures to Generalisation Operators. International Journal of Geographical Information Science, 22 (1-2) 133-165.

Newcomer, E., Lomow, G. 2004. Understanding SOA with Web Services. Addison-Wesley Professional.

NFP/Eionet. 2007. Preparation document for a Commission Communication on SEIS and related matters; NFP/Eionet group meeting Doc: EEA/NFP (2007) 02 - 2 Item: 2.1.1 8 May 2007

Nogueras-Iso, J., F.J. Zarazaga-Soria, R. Béjar, P.J. Álvarez, P.R. Muro-Medrano, 2005. OGC Catalog Services: a Key element for the development of Spatial Data Infrastructures. Computers and Geosciences, 31 (2) 199-209

Official Journal of the European Union, 2003. Directive 2003/ 98/EC of the European Parliament and of the Council of 17 November 2003 on the re-use of public sector information. L 345, 2003, pp. 0090–0096.

Olaya, V., 2007, SEXTANTE: a gvSIG-based platform for geographical analysis, FOSS4G 2007, Available at http://www.foss4g2007.org/presentations/view.php?abstract_id=123 (accessed January 28, 2009)

ORCHESTRA Consortium. 2008. Orchestra, an open service architecture for risk management. http://www.eu-orchestra.org/docs/ORCHESTRA-Book.pdf

Papazoglou, M. P., Yang, J., 2002. Design Methodology for Web Services and Business Processes. In Proc. of Workshop on Technologies for E-Services (TES 2002). Springer, LNCS 2444, 54-64.

Papazoglou, M., 2007. Web Services: Principles and Technology. Pearson - Prentice Hall.

Papazoglou, M., Heuvel, W-J van den, 2007. Service oriented architectures: approaches, technologies and research issues. The VLDB Journal 16 (3) 389-415.

Pecar-Ilic, J., Ruzic, I., 2006. Application of GIS and Web technologies for Danube waterway data management in Croatia. Environmental Modelling & Software 21 (11) 1562-1571

Percival, G. 2006. OpenGIS International Standards for GEOSS Interoperability Arrangements. IEEE International Conference on Geoscience and Remote Sensing Symposium. IGARSS 2006. 2481-2484

Percival, G. (ed). 2008. OGC Reference Model. Open Geospatial Consortium Inc. Date: 2008-11-11. Reference number: OGC 08-062r4. Version: 2.0

Petrie, C., Bussler, C., 2008. The Myth of Open Web Services: The Rise of the Service Parks. IEEE Internet Computing 12 (3) 94-96.

Portele, C. 2007. OpenGIS Geographic Markup Language (GML) Encoding Standard. http://www.opengeospatial.org/ standards/gml

Ramamurthy, M. K. 2006. A new generation of cyberinfrastructure and data services for earth system science education and research. Advances in Geo-Science.

Rajabifard, A., Williamson, I. P. 2001. Spatial Data Infrastructures: Cocept, SDI Hierarchy and Future Directions. Proceedings Geomatics 2001 (Australia). Available on-line at: http://www.sli.unimelb.edu.au/research/publications/IPW/4_01Raj_Iran.pdf

Rajabifard, A., Feeney, M-E. F., Williamson, I. P. 2002. Future directions for SDI development. International Journal of Applied Earth Observation and Geoinformation 4 11–22

Rajabifard, A., Binns, A. and Williamson, I. 2005. Administering the Marine Environment – the Spatial Dimension, Journal of Spatial Science, 50 (2) 69-78

Rajabifard, A., Kalantari, M., Binns, A. 2009. SDI and Metadata Entry and Updating Tools. SDI Convergence. Research, Emerging Trends, and Critical Assessment. B. van Loenen, J.W.J. Besemer, J.A. Zevenbergen (Editors). Nederlandse Commissie voor Geodesie Netherlands Geodetic Commission 48, 2009.

Reitsma, F., Albrecht, J., 2005. Modeling with the Semantic Web in the Geosciences. IEEE Intelligent Systems 20 (2) 86-88.

Reference Model for the ORCHESTRA Architecture (RM-OA) V2 Rev 2.1 as OGC Best-practice document 07-097. http://portal.opengeospatial.org/files/?artifact_id=23286

Rocha, A, Cestnik, B., Oliveira, M.A. 2005. Interoperable Geographic Information Services to Support Crisis Management in J. Li and C.Vangenot (Eds): 5th International Workshop on Web and Wireless Geographical Information Systems, LNCS 3822 Springer-Verlag. Laussane, Switzerland, 246-255

Rohner, N., Schrogl, K-U, Cheli, S. 2007. Making GMES better known: Challenges and opportunities. Space Policy 23 (4) 195-198

Rutledge, G. K., Alpert, J., Stouffer, R. J., and Lawrence, B. 2002. The NOAA Operational Archive and Distribution System (NOMADS), Proceedings of the Tenth ECMWF Workshop on the use of High Performance Computing in Meteorology, edited by: Zwiefhofer, W. and Kreitz, N., World Scientific, 106–129

San-Miguel-Ayanz, J. 2008. The European Forest Data Center (EFDAC) - on going initiatives on data dissemination at the EU level - Forest C&I Analytical Framework and Report Workshop. Finnish Forest Research Institute, Joensuu, Finland, May 19-21

Sanz, J., Montesinos, M. 2009. Current Panorama of the FOSS4G Ecosystem," The European Journal for the Informatics Professional, X 43-51.

Shneiderman, B., 2007. Web Science: A Provocative Invitation to Computer Science. Communication of ACM 50 (6) 25-27.

Scholten, H., Fruijter, S., Dilo, A., van Borkulo, E. 2008. Spatial Data Infrastructure for Emergency Response in Netherlands. Remote sensing and GIS technologies for monitoring and prediction of disasters. Nayak and Zlatanova Eds, 179-197

Scholten, M., Klamma, R., Kiehle, C. 2006 Evaluating Performance in Spatial Data Infrastructures for Geoprocessing," IEEE Internet Computing, 10 (5) 34-41

Sheth, A. P. 1999. Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics. Interoperating Geographic Information Systems. M. Goodchild, M. Egenhofer, R. Fegeas and C. Kottman, Kluwer Publisher: 5--30.

Shen, Z., Ming, D and Li, J. 2005. Remotely sensed image distributed procesing system design with Web Services technology, in Proceedings of the IEEE IGARSS , pp. 4244-4247

Smits, P. C., Friis-Christensen, A., 2007. Resource Discovery in a European Spatial Data Infrastructure. IEEE Transactions on Knowledge and Data Engineering 19 (1) 85-95.

Soh L-K, Zhang J, Samal A. 2006. A Task-Based Approach to User Interface Design for a Web-Based Hydrologic Information Systems. Transactions in GIS 10 (3) 417–449

Schut, P. (ed). 2007. OGC Web Processing Service (WPS) version 1.0.0. OGC Standard Document, Open Geospatial Consortium.

Teng, J., Vaze, J., Tuteja, N. K., Gallant, J.C., 2008. A GIS-Based Tool for Spatial and Distributed Hydrological Modelling: CLASS Spatial Analyst. Transactions in GIS 12 (2) 209-225.

Turner, A. 2006. Introduction to Neogeography. CA: O'Reilly Media Inc

Tulloch, D. L. 2007. Many many maps: Empowerment and online participatory mapping. First Monday.

United Nations. 2002. Plan of Implementation of the World Summit on Sustainable Development. Johannesburg (South Africa), August 26 to September 4, 2002. <http:/ /www.un.org/esa/sustdev/documents/WSSD_POI_PD/English/POIToc.htm>.

Peter van Oosterom and Sisi Zlatanova (editors). 2008.Creating Spatial Information Infrastructures: Towards the Spatial Semantic Web. CRC Press: Boca Raton, Florida, 2008. xxiii and 185 pp., index ISBN-10: 1420070681; ISBN-13: 978-1420070682

UK Ministry of Defence (MOD), 2005, Networked Enabled Capability JSP 777 Edition 1, UK Ministry of Defence

Vandenbroucke, D., Crompvoets, J., Vancauwenberghe, G., Dessers, E., Van Orshoven, J. 2009. A Network Perspective on Spatial Data Infrastructures: Application to the Sub-national SDI of Flanders (Belgium). Transactions in GIS, 13(1) 105–122

Vicente-López, F.J. & Poyatos-Hernandez, C., 2007. IDE y geoportales aplicados a los incendios forestales: SIGIF, el caso de la Comunidad Valenciana. En Sevilla, Spain. Available at: http://www.fire.uni-freiburg.de/sevilla-2007/contributions/doc/SESIONES_TEMATICAS/ST4/deVicente_Poyatos_SPAIN_Valencia.pdf.

Voisard, A., Schweppe, H. 1998. Abstraction and decomposition in interoperable GIS. International Journal of Geographical Information Science 12(4) 315—333

Vretanos, P.A. 2005. OGC Web Feature Service Implementation Specification, Version 1.1.0. Open Geospatial Consortium

Weaver, J. L. , Gao, W., Chin, S., Iverson, D. 2009. Pro JavaFX Platform: Script, Desktop and Mobile RIA with Java Technology , Apress, 2009

Whiteside, A. (ed.). 2005. OGC Web Services Common Specification Version 1.0. OGC Implementation Specification, Open Geospatial Consortium

Whiteside, A., Evans, J.D. (eds). 2008. Web Coverage Service (WCS) Implementation Standard. Open Geospatial Consortium

Wiederhold, G. 1992. Mediators in the Architecture of Future Information Systems. IEEE Computer Magazine 25(3) 38--49.

Williamson, I.P. 2001 Land Administration "Best Practice" - providing the infrastructure for land policy implementation. Journal of Land Use Policy 18(4) 297-307

Williamson, I.P. 2003. Chapter 1: SDIs: Setting the scene. In Developing spatial data infrastructures: From concept to reality, eds. I. Williamson, A. Rajabifard, and M.-E. F. Feeney, 3–16. London: Taylor & Francis.

World Wide Web Consortium. 2004. Web Services Architecture W3C Working Group Note (Feb. 11 2004). W3C. www.w3.org/TR/2004/NOTE-ws-arch-20040211/.

Wytzisk, A, Simonis, I., Raape, U. 2003. Integration of HLA Simulation into a Standardized Web Service World. European Simulation Interoperability Workshop 2003, Stockholm, Sweden.

Yang, C., Li, W., Xie, J., Zhou, B. 2008.Distributed geospatial information processing: sharing distributed geospatial resources to support Digital Earth. International Journal of Digital Earth, 1(3) 259-278

Yue, P., Di, L., Yang, W., Yu, G., Zhao, P., 2007. Semantics-based automatic composition of geospatial Web service chains. Computers & Geosciences 33 (5) 649-665.

Zlatanova, S., van Oosterorn, P., Verbree, E. 2006. GEO-INFORMATION SUPPORT IN MANAGEMENT OF URBAN DISASTERS. Open house international 31(1)

Zlatanova, S., Fabbri, A.G. 2009, Geo-ICT for Risk and Disaster Management, in Scholten, v/d Velde & van Manen (eds): Geospatial Technology and the Role of Locations in science, Springer Dordrecht, 239-266

Zlatanova, S., Dilo, A. 2010. A DATA MODEL FOR OPERATIONAL AND SITUATIONAL INFORMATION IN EMERGENCY RESPONSE: THE DUTCH CASE Proceedings of Gi4DM 2010 Conference on Geomatics for Crisis Management. Torino, Italy, Feb 2010.

Annex A: Aware Processing Services Design

Data Conversion WPS

The DataConversion WPS service contains a set of processes with functionality regarding data format conversion and coordinates transformation. Next we describe the abstract specification of this service and its operations, in a platform independent way, to be easily mapped in the implementation phase with a chosen technology. Table 3 shows the processes implemented.

Name	DataConversion
Description	This WPS contains a set of data format processing algorithms
Processes	Shp2GML, CoordTransformer

Table 3. DataConversionWPS service overview

Figure 75. DataConversion conceptualization shows the UML diagram with the interface and functional external behaviour o this service.

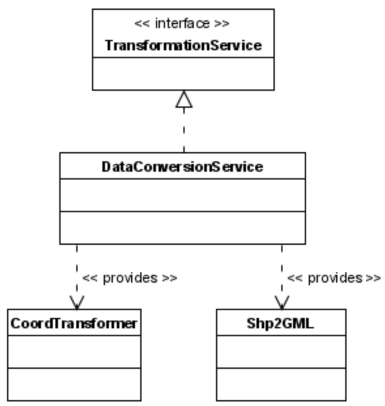

Figure 75. DataConversion conceptualization

Data Conversion WPS - Shp2GML Process

The Shp2GML process transforms a given ESRI shapefile to GMLformat. The ESRI Shapefile is a geospatial vector data format. As we can see in the UML diagram, this process receives an input which is the URI to the shapefile and will return one output is the same vector data but encoded using OGC GML format which is an XML-based format. The data type FeatureCollection is a collection of vector entities.

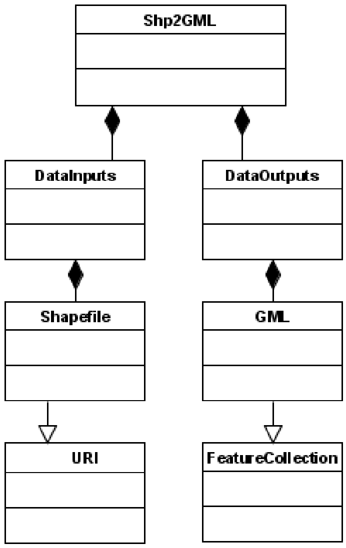

Figure 76. Shp2GML abstract specification

Topology WPS

This WPS contains a set of processes concerned about topology relationship computations and spatial operations. Table 4 shows the functional definition and the available processes.

Name	TopologyWPS
Description	This WPS contains a set of topology relationship computations as the case of those defined in the OGC Simple Feature Specification. Other spatial operations not included in this OGC specification may be defined as well
Processes	Area, Intersection, Buffer,

	MaxExtent SnowPercentage, GetFeatureByAttribute, Thiessen

Table 4. Topology WPS service overview

Figure 77 is the UML diagram that shows processes offered by this service.

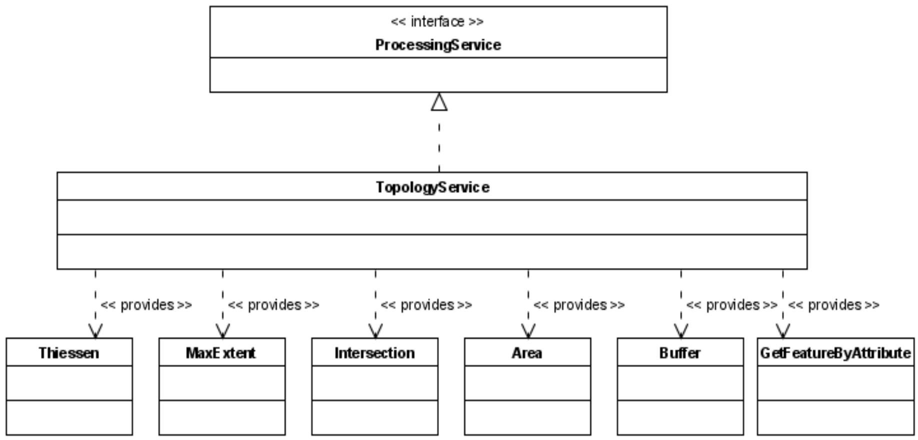

Figure 77. Conceptualization of TopologyWPS

Topology WPS** - **Area Process

It calculates the total area of the input geometry. The Figure 78 is the UML diagram of this process where we can appreciate inputs outputs to be sent in order to execute this process. The input is FeatureCollection and the output is a literal number with the area of the geometry.

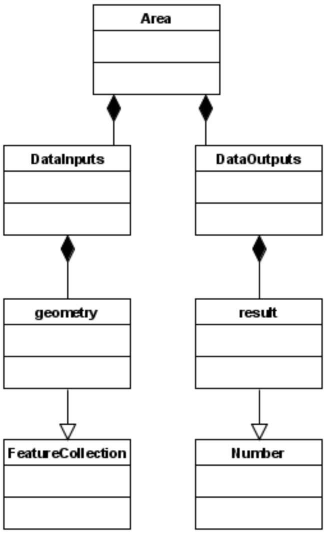

Figure 78. Area process abstract specification

Topology WPS* - *Intersection Process

Intersection takes as input two geometries and returns the intersection. The intersection of two geometries A and B is the set of all points which lie in both A and B, as we see in Figure 79. Figure 80 shows the inputs type is FeatureCollection with the entities to be intersected and the output is another FeatureCollection with the intersection.

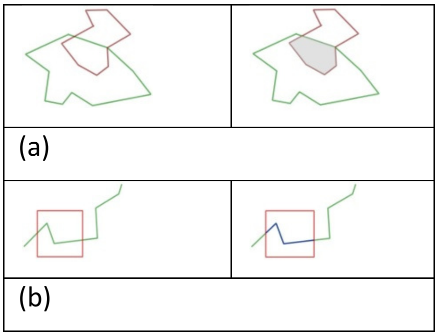

Figure 79. Intersection operation. (a) polygon-polygon; (b) polygon-line

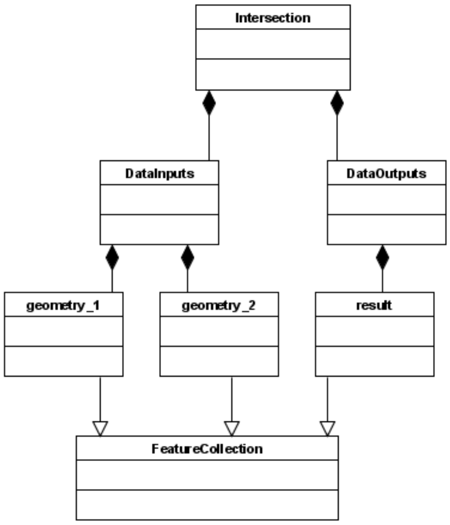

Figure 80. Intersection process abstract specification

Topology WPS - Buffer Process

SimpleBuffer takes a geometry and returns a new geometry widened to a specific width. Figure 81 illustrates some examples of the buffer operation. The associated UML diagram is shown in Figure 82.

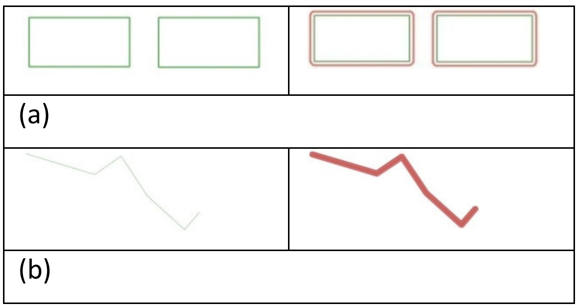

Figure 81. Buffer operation. (a) polygon; (b) line

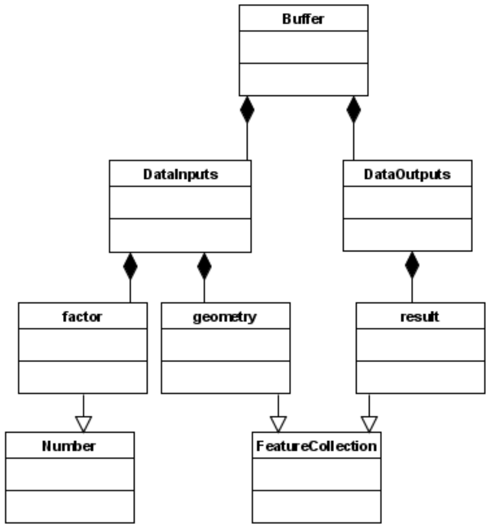

Figure 82. Buffer process abstract specification

Topology WPS - MaxExtent Process

MaxExtent calculates a minimum bounding rectangle (bounding box) as an expression of the maximum extents of a given polygon. Some graphical examples are shown in Figure 83. The associated UML diagram is shown in Figure 84.The input type is FeatureCollection and the output is a geometry of box type.

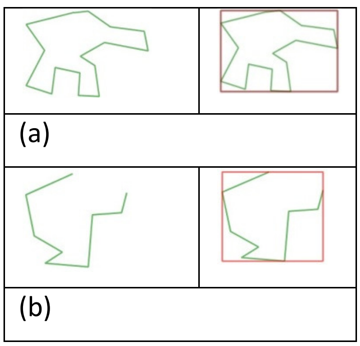

Figure 83. MaxExtent operation. (a) polygon; (b) line

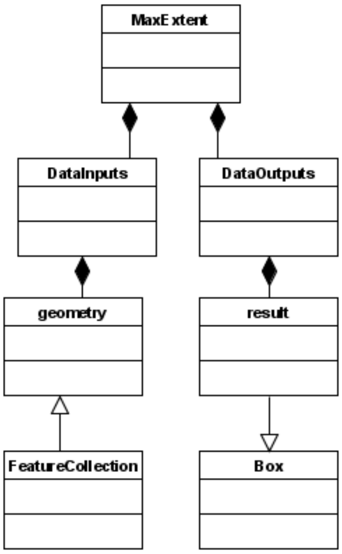

Figure 84. MaxExtent process abstract specification

Topology WPS - Thiessen Process

This process calculates the Thiessen diagram which is a special kind of decomposition of a metric space determined by distances to a specified discrete set of objects in the space, e.g., by a discrete set of points. Figure 85illustrates the Thiessen diagram in the common case of the plane. In this example, a set of points S is given, and the Voronoi diagram for S is the partition of the plane which associates a region V(p) with each point p from S in such a way that all points in V(p) are closer to p than to any other point in S.

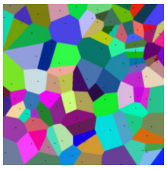

Figure 85. Thiessen polygons

Figure 86 is the UML diagram. The first input which is of box type is split according with the list of point given by the second input as a FeatureCollection, this process returns a list of polygons.

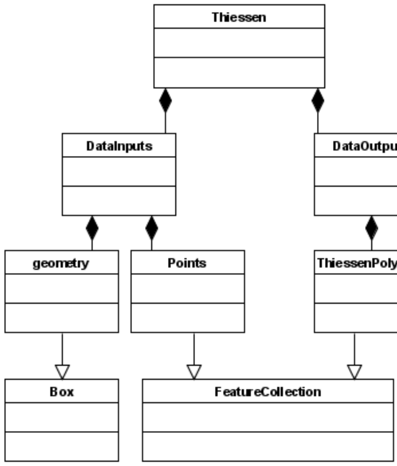

Figure 86.Thiessen process abstract specification

Topology WPS - SnowPercentage Process

SnowPercentage calculates the percentage of snow that covers certain elevation zone of the study basin. The figure below is the UML diagram.

The inputs are the Snow cover area (SCA) in vector format and the region of a basin where we want to calculate the percentage of snow. Usually this region is an elevation zone defined by a FeatureCollection and the output is a literal number with the percentage of snow covering this basin region.

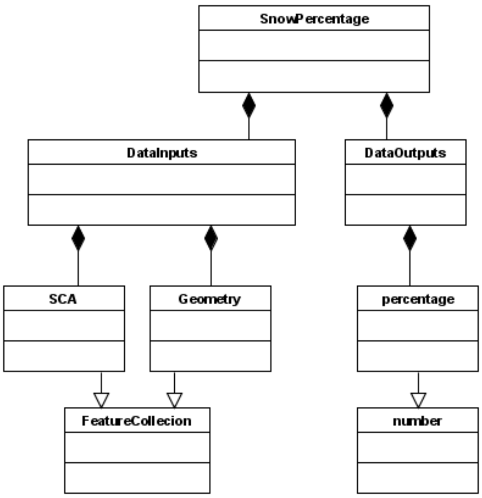

Figure 87. SnowPercentage process abstract specification

This process processes the data by performing a service chain: SnowPercentage executes first the area process of one elevation zone. Secondly, it performs a call to the AWARE WFS serving the SCA of the basin region at a certain date. After this, the Intersection process is called to obtain the intersected snow features with the current elevation zone. Finally the SnowPercentage process invokes the Area process, this time to calculate the area of this intersected region. The SnowPercentage process will then return a snow percentage (snow-covered area divided by elevation zone area) of a certain elevation zone in a concrete day.

Sextante WPS

This WPS obtains its name from the software Sextante (SEXTANTE), a toolset with more than 200 functions for raster and vector data processing. From all the of the SEXTANTE functionality we have chosen the methods that are interesting for our user requirements. Table 5 summarizes the main features of the SextanteWPS module.

Name	Sextante WPS
Description	This WPS contains a set of raster image processing algorithms
Processes	CoordinateElevation StationElevation HypsometricElevation ElevationZones ElevationCurves Reclassify Vectorize

Table 5. Sextante WPS service overview

Figure 88 is the UML diagram with the processes offered by this service.

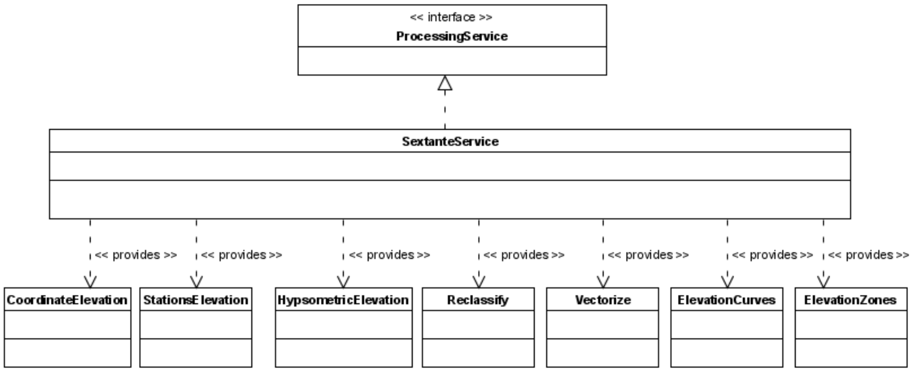

Figure 88. Conceptualization of SextanteWPS

Sextante WPS - CoordinateElevation Process

The CoordinateElevation process returns the elevation value on the DEM at a given coordinate point. The Figure 89 shows the UML diagram of this process.

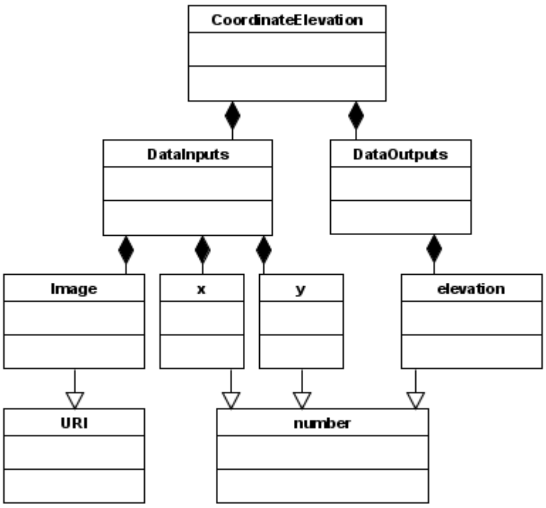

Figure 89. CoordinateElevation process abstract specification

Sextante WPS - HypsometricElevation Process

This process calculates the hypsometric elevation of an elevation zone delimited by an elevation range. Figure 90 shows the UML diagram of this process.

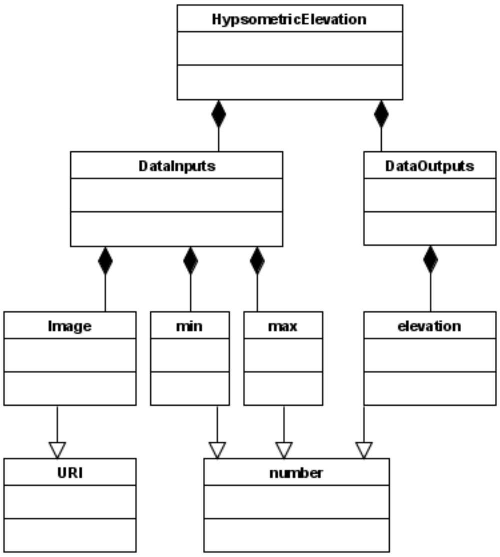

Figure 90. HypsometricElevation process abstract specification

Sextante WPS - StationsElevation Process

This process returns the elevation of a given list of coordinates. Figure 91 shows the UML diagram of this process.

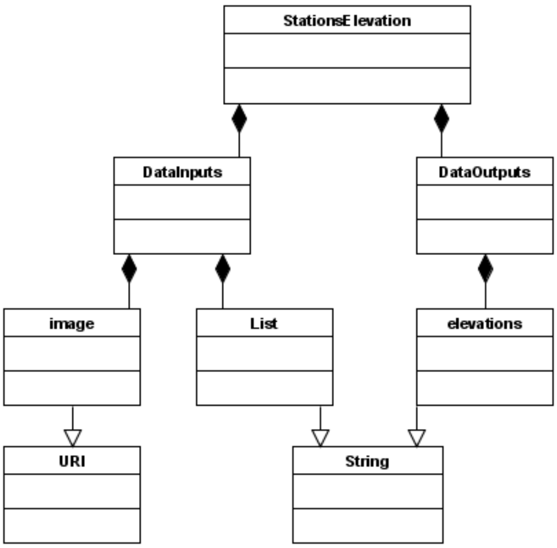

Figure 91. StationsElevation process abstract specification

Sextante WPS - Reclassify Process

The Reclassify process is a raster algorithm for classification. For example consider a satellite image with temperature values of a certain terrain. This image is a raster file where each cell represents a temperature value, these values can be classified in a number of ranges, in Figure 92 it is classified in three different values according to three ranges. Figure 93 shows the UML diagram of this process.

15	17	20	21	23
16	18	22	23	24
19	19	22	23	24
23	23	27	28	28
22	22	31	32	33

1	1	2	2	2
1	1	2	2	2
1	1	2	2	2
2	2	3	3	3
2	2	3	3	3

LEGEND

1 = "15-20"
2 = "20-25"
3 = "25-35"

Figure 92. Classify algorithm according to three temperature ranges

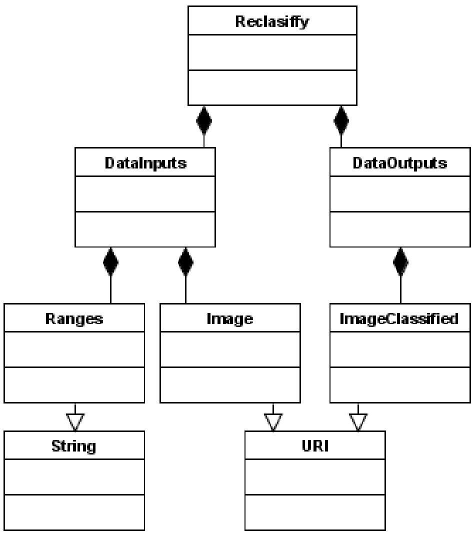

Figure 93. Reclassify process abstract specification

Sextante WPS - Vectorize Process

The Vectorize process transforms a DEM file which has been classified into vector data encoded with chosen vector format. Figure 94 shows the UML diagram of this process.

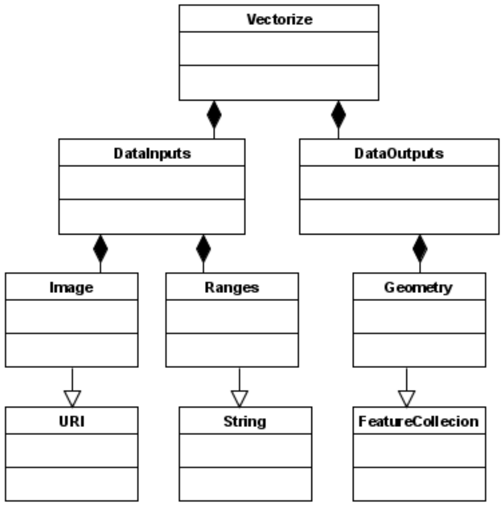

Figure 94. Vectorize process abstract specification

Sextante WPS - ElevationZones Process

This process analyzes an input DEM file and extracts the cells in the DEM belonging to that desired elevation zone (reclassify), and then vectorize this zone returning the polygons defining this area a vector format. This process invokes two processes according to a predefined chain. This is possible only because we know a priori the input and output parameters of the service processes invoked. Figure 95 shows the UML diagram of this process.

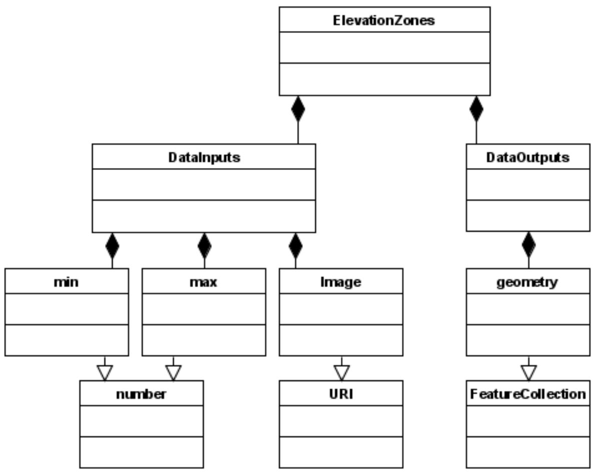

Figure 95. ElevationZones process abstract specification

The execution is like in the figure below, taking a DEM as input and some ranges, the elevation zones process invokes the reclassify process, obtaining a classified image, and send this output as an entry to the vectorize process, which returns the elevation zones as a feature collection in vector format.

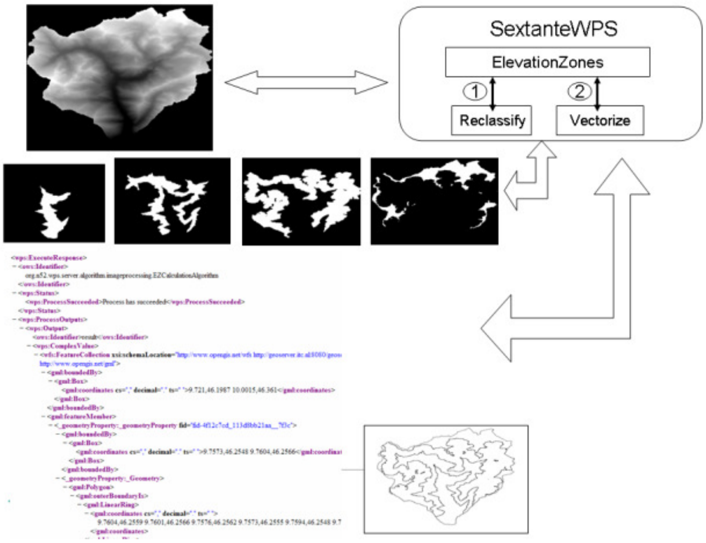

Figure 96. Elevation Zones functional chain

SRMIDL WPS

The SRMIDL WPS service solely contains a process to invoke IDL-based routines within the context of the SRM model. In this way, it is possible to run IDL routines in a remote, distributed manner. Table 6 summarizes the main characteristics of the SRMIDLWPS service.

Name	SRMIDL WPS
Description	This WPS contains processes to run SRM model implemented as an IDL routine
Processes	SrmIdl

Table 6. SRMIDLWPS service overview

Figure 97 shows the UML diagram that shows the processes offered by this service.

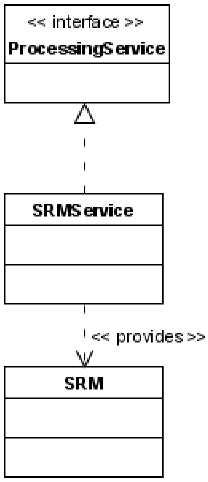

Figure 97. SRMIDLWPS conceptualization

SRMIDL WPS - SRM Process

The SRM process runs IDL routines of SRM model. In particular, it runs two IDL routines: the first one is a routine to perform temporal interpolation for the depletion curves; the other is to calibrate the model parameters. Figure 98 is the UML diagram of this process.

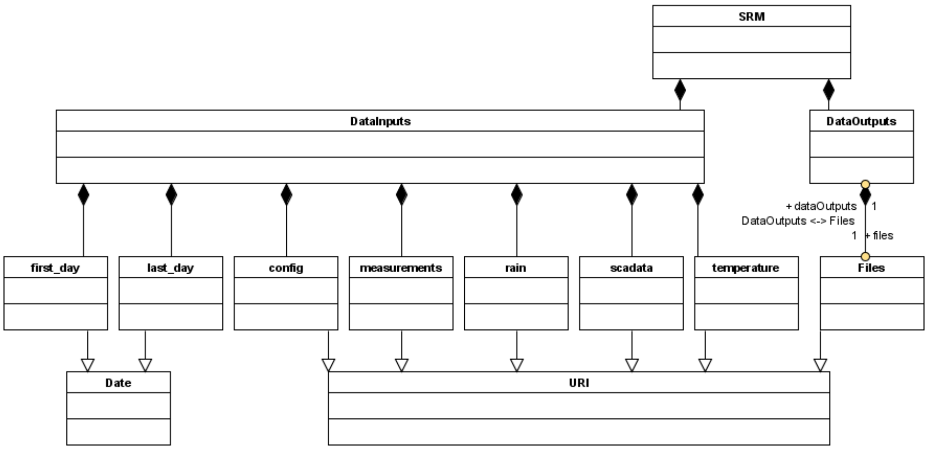

Figure 98. SRMIDL process abstract specification

AWARE ViewChart Services

Diagram services are not included in the first draft of the INSPIRE Service Network as a functionality of an identified service type, but it is an important requirement for AWARE users, who need to be able to represent information, not as maps, but as descriptive plots showing information in a graphical way. Together with the hydrologists team we analyzed and defined all the plot creation requirements of the AWARE application. The resulting processes have been grouped in a Service called ChartWPS.

Chart WPS

The Chart WPS module contains a set of processes providing functionality for chart and plots creation. The different processes mainly render an image (in png format) containing the plot returning the URL to the resulting image. Table 7 summarizes the main features of the Chart WPS module. Next, we describe the abstract specification of this service and its operations, in a platform independent way, to be easily mapped in the implementation phase to the chosen platform.

Name	ChartWPS
Description	This WPS contains chart creation functionality
Processes	DepletionCurvePlot DischargePlot HbvRunoffPlot HbvSWEPlot SensorDataChart

Table 7. chartWPS service overview

Figure 99. ChartWPS conceptualizationis the UML diagram that shows the external behaviour of this service, i.e., the processes offered by this service.

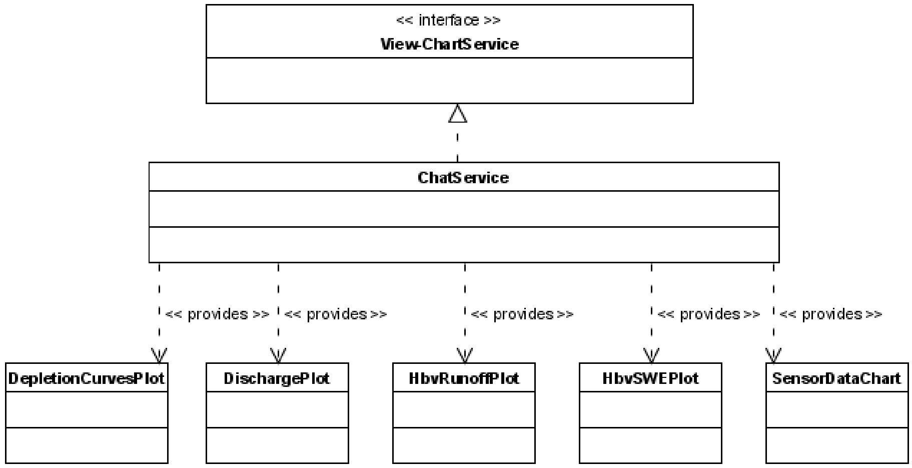

Figure 99. ChartWPS conceptualization

Chart WPS - Depletion Curves Plot Process

This process creates a depletion curves plot for the SRM model given series of values as illustrated in Figure 100.

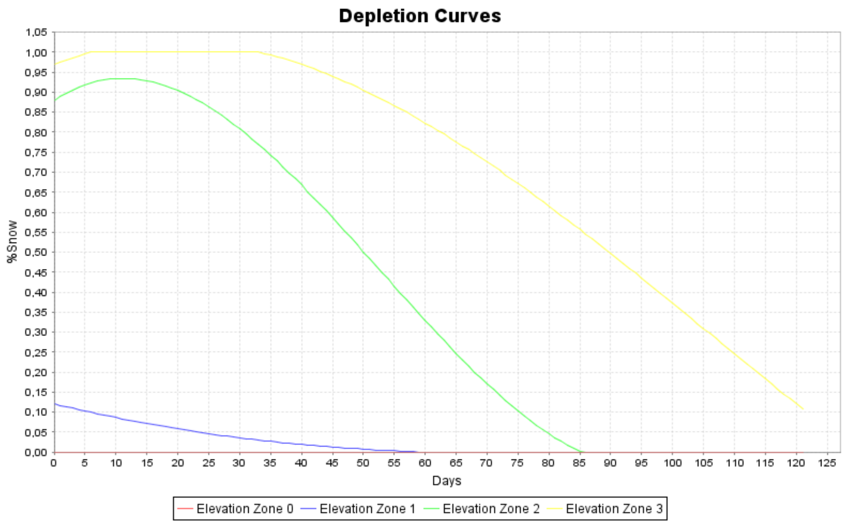

Figure 100. SRM Depletion Curves Plot

Figure 101 shows the UML diagram of this process.

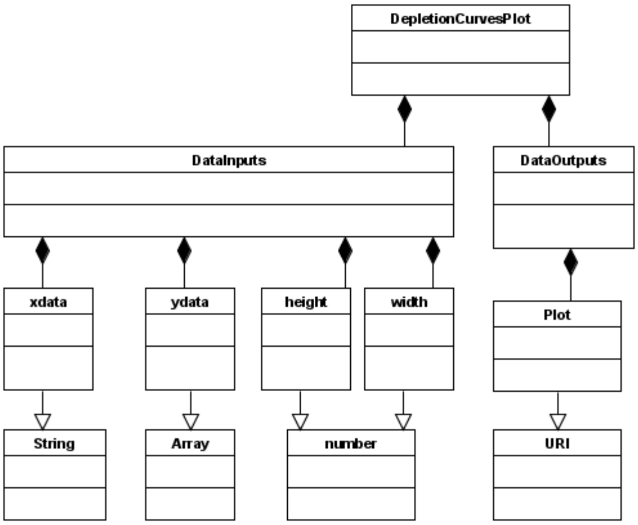

Figure 101. DepletionCurvesPlot process abstract specification

Chart WPS - Discharge Plot Process

This process creates a discharge plot for the SRM model given series of values as illustrated in Figure 102. This plot compares in two plots the real discharge with calibrated and simulated discharge along the time axis.

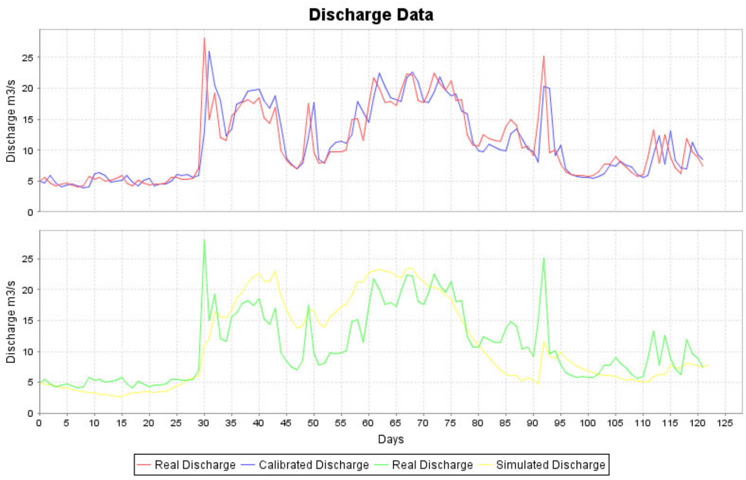

Figure 102. SRM Discharge Curves plot

Figure 103 shows the UML diagram of this process with the input and output parameter types.

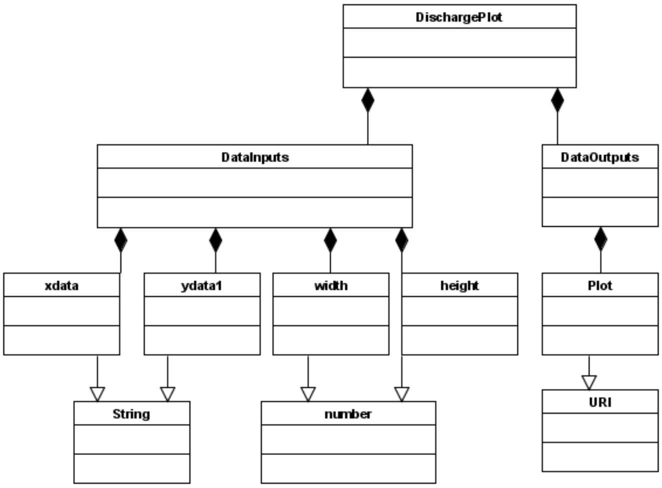

Figure 103. DischargePlot process abstract specification

Chart WPS - HBV Runoff Plot Process

This process creates a runoff plot for the HBV model given series of values as illustrated in Figure 104. It compares the real precipitation data provided by the meteorological stations with the simulated and observed discharge values along the time axis.

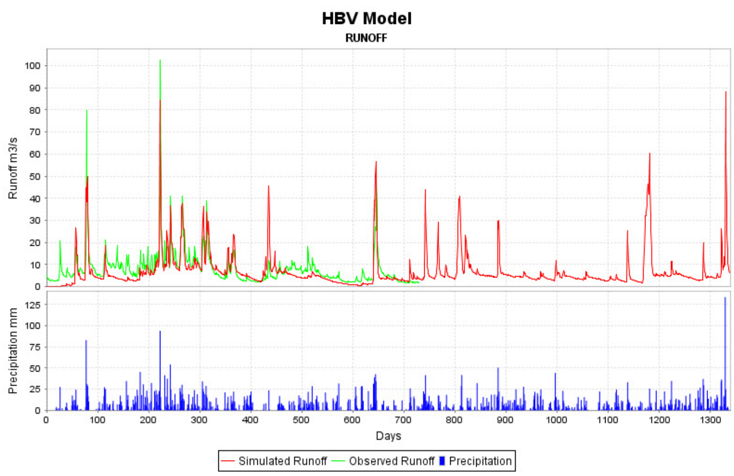

Figure 104. HBV runoff plot

Figure 105 shows the UML diagram of this process.

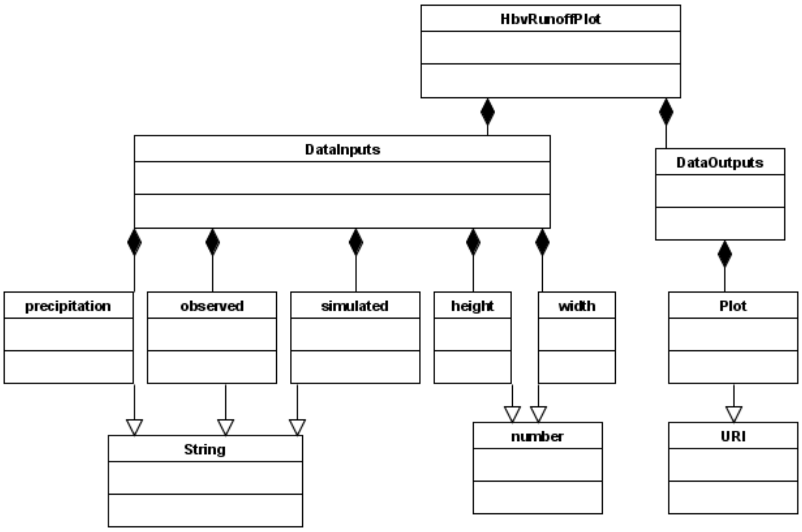

Figure 105. HBVRunoffPlot process abstract specification

Chart WPS - HBV SWE Plot Process

This process creates a SWE plot for the HBV model given series of values as illustrated in Figure 106. This plot compares de Snow-Water-Equivalent (SWE) value with the EO data, i.e., the data coming from satellite images.

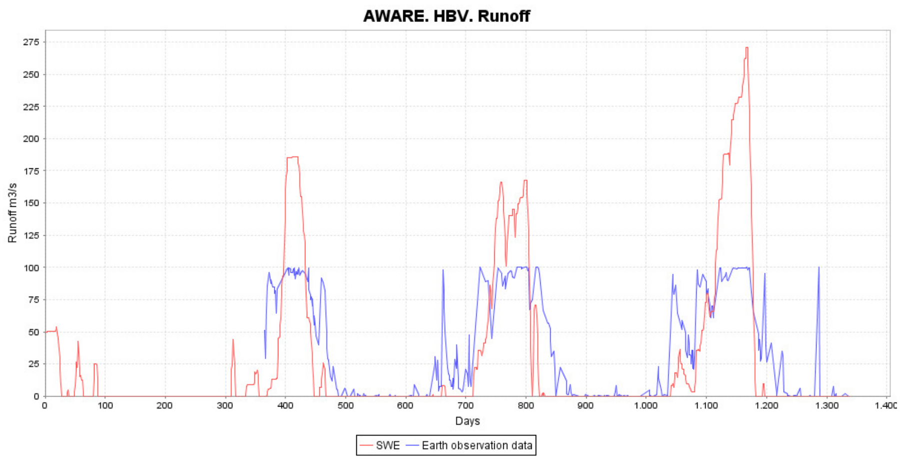

Figure 106. HBV Snow-Water-Equivalent plot

Chart WPS - SensorData Process

This process creates a plot with the data of a given sensor. This is used to represent the list of values of a Rain gauge and/or a thermometer in a Weather station. The plot is a single line along the time axis. The UML diagram of this process is shown in the Figure 107.

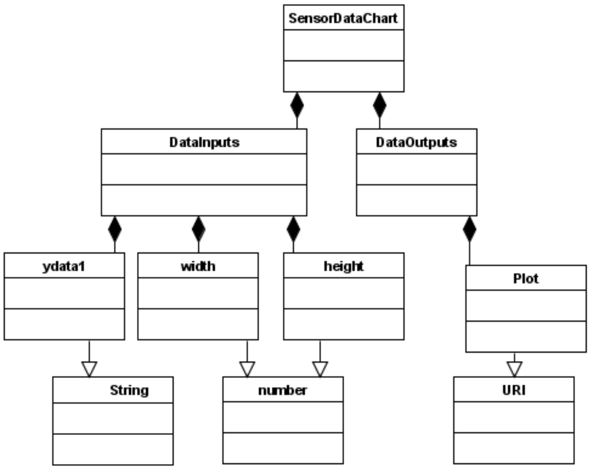

Figure 107. SensorData process abstract specification

Data Model

The data and metadata specification by the data Drafting Team in INSPIRE is still in its development phase. These specifications integrate the definition of mandatory themes in fields that traditionally manage geospatial information. The definition includes specifications of data and metadata schemas. Hydrology data specification will be drafted by a community of hydrological specialists. Assuming they use compatible XML schemas, they should be compatible with our services.

Although it is out of the scope of this thesis work we define briefly only one example of data specification, the application schema of two of the entities of the information model we have used to model and handle the geospatial information needed in the AWARE application. Some of the processing services transform raster data to generate or create new vector data. In order to have an information model to manage this information, we have defined the following entities:

Basin Feature

The basin model extends the simple feature model. A drainage basin is an extent of land where water from rain or snowmelt drains downhill into a body of water, such as a river. Figure 108, shows a basin represented with raster data, concretely a DEM file, and vector data which in this example is a polygon with the basin boundary.

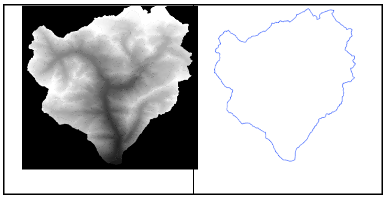

Figure 108. Basin example in raster and vector format

The next figure, shows the abstract basin Model, which we modelled as an entity having a Multipolygon and an attribute which is the basin identifier.

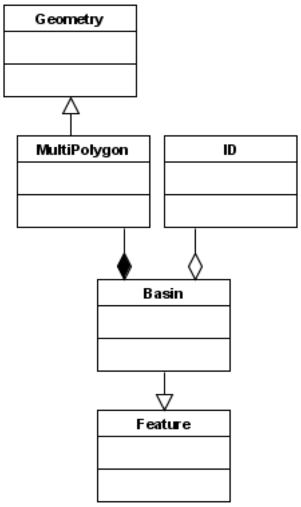

Figure 109. Basin Abstract model

Elevation Zone Feature

The elevation zone model will extend the simple feature model. An elevation zone of a terrain is the region comprised in an elevation range. Figure 111 shows the abstract elevation zone feature.

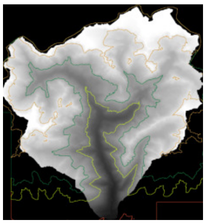

Figure 110. Example of a drainage basin divided in 4 elevation zones

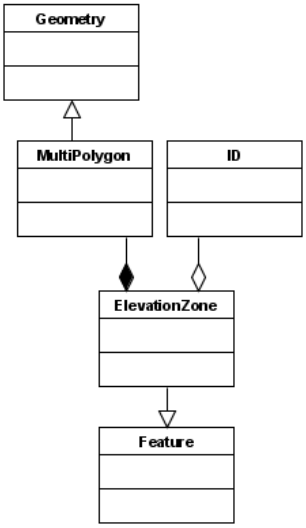

Figure 111. Elevation Zone Abstract model

Annex B: Aware Processing Services Implementation

DataConversion WPS

This service has been implemented based on 52North, like all the processing services we will explain in detail in this section. We explained in the design section the purpose and functionality of DataConversionWPS, we describe here the implementation decisions, and the open source libraries we use for the implementation of the processes. To implement the processing algorithms o processes, we used Geotools API (Geotools) and JTS (Java Topology Suite) libraries.

Shp2GML Process

This process transforms a file in ESRI Shapefile format a GML version 2.1.2 format. GML provides the basis for domain- or community-specific "Application Schemas", which in turn support data interoperability within a community of interest. In the process there is an input parameter which determines the application schema to be use to generate the GML. For instance, in the AWARE project, this process is called with the basin schema (see Data Model section) when a shapefile of a drainage basin has to be converted to GML.

CoordTransformer Process

This process transforms the coordinates in a GML file from one coordinate reference system to another.

Topology WPS

We explained in the design section the purpose of TopologyWPS, now we describe the implementation decisions, and the libraries we have use for the implementation of the processes. We have used Geotools API and JTS libraries. To generate the Thiessen polygons in the Thiessen algorithm we adapted the open source applet provided by Paul Chew (Chew).

Area Process

The area process calculates the area of a FeatureCollection which is a list of geometries with attributes. The input parameter, the feature collection, must be in GML format, and the result is a literal numeric value.

Intersection Process

The intersection process calculates the intersected geometry of two FeatureCollection. The input parameters are two feature collections that must be in GML format, and the result is another feature collection with all the intersected geometries. This process does not reproject the data, so the execute request must contain the geometries the same SRS, otherwise the intersection will be an empty collection

Buffer Process

The buffer process calculates the buffered geometry of a FeatureCollection given a positive width. The input parameters are a feature collection in GML format, and the factor to use to calculate the buffered geometry which is a literal numeric positive value. The output is the buffered geometry in GML format.

MaxExtent Process

The maxExtent process calculates the minimum box, a rectangle given by minimum x and y coordinates and maximum x and y coordinates, that covers the input geometry, to implement it. The input parameter is a feature collection in GML format, and the output is a box geometry in GML format.

GetFeatureByAttribute Process

The GetFeatureByAttribute selects a group of features from a feature collection that fulfil a condition for a given attribute. The input parameter is a feature collection in GML format, and two literal values containing the attribute name and value. The schema and the srs (reference system) inputs are necessary to generate the output GML. The output is a feature collection with the selected features in GML format.

Thiessen Process

The thiessen process calculates the thiessen polygons of a box, a rectangle given by minimum x and y coordinates and maximum x and y coordinates, and a list of points, the box and the points will be pre-formatted strings, coordinates comma separated. To implement it we have made use of an open source developed by Paul Chew (Chew) that we have modified to work with Geotools library and JTS. The SRS and shema are needed to generate a proper GML as output, to generate it we use the application schema designed for thiessen polygons that is explained in the Data Model section.

SnowPercentage Process

The SnowPercentage process calculates the percentage of snow that is covering a region. This is a domain specific process but we considered reusable and important enough to be expose as a process. Internally this process in a chain of other processes acting as an opaque-chaining engine. The inputs are a snow cover area file, which is vector data of the polygons of snow in a certain area. In the framework of the AWARE project, these data is normally a Web Feature Service (Feature access or download service type) request. The second input parameter is the region that we want to calculate the percentage of snow, in AWARE normally this region is an elevation zone of the drainage basin defined as a polygon collection, this data like the SCA is in GML format. This process does not reproject the data, so the execute request must contain the SCA and the region in the same SRS. In de design section we did explain how the chain is performed by this process.

SextanteWPS

The implementation of this service is based on 52North as well. For the implementation of its processes we used SEXTANTE. It includes extensions for hydrological and geomorphometric analysis, among many others, and also basic raster and vector data handling tools. We have design this WPS to maximize interoperability and reusability. And to implement it we follow a wrapping methodology using the Adapter pattern to easily migrate exiting functionality to WPS.

SEXTANTE is written in Java, so we can add, reuse and modify the code including the parts that we need in our 52North framework. To integrate SEXTANTE into the 52N WPS framework in a way that allows the easy addition of functionality to SEXTANTE when needed, we have followed the Adapter pattern (Gamma, 95), as illustrated in Chapter 4 and 5. As previously mentioned the SEXTANTE structure splits the logic from the graphical interface, thus facilitating integration. To lighten the WPS we remove the GUI components from SEXTANTE. Currently, SextanteWPS provides several SEXTANTE processes such as Reclassify and Vectorize. Besides these processes, we have added new ones that integrate multiple SEXTANTE algorithms, as the case of the complex process ElevationZones.

CoordinateElevation Process

This process returns the elevation of a point given an x, y coordinates. This process like the rest of the processes in this Service are implemented to extract information of a Digital Elevation Model (DEM) which is a raster file containing for each cell the average value of altitude for the terrain portion contained in the grid cell. The tests of these processes have been done with DEM files with a 20x20m resolution in GeoTiff[xxxviii] format.

Hypsometric Elevation

This process calculates the hypsometric elevation of an elevation zone of a drainage basin. As input parameters we have the URI to a DEM file, the min and max numbers defining the elevation interval of the elevation zone. The process calculates the average elevation which will be the output parameter as a numeric value.

Stations Elevation

This process calculates the elevation of list of meteorological stations, represented by a list of points. As input parameters we have the URI to a DEM file, the list(comma-separated) of the coordinates corresponding to the stations. This process reads the value of the cell corresponding to the coordinates of each station and returns these elevations as a string in a comma –separated format.

Reclassify Process

This process performs a classification depending on elevations. As input parameters we have the URI to a DEM file, the second input are the categories (elevation ranges). This process using the adapter pattern calls a SEXTANTE process which is reclassifying the DEM file. The output is a URL to the reclassified image.

Vectorize Process

This process performs a vectorization of a classified image. As input parameters we have the URI to a classified image in Geotiff format. Since this process will generate vector data in GML format, we need as input parameters the SRS and the information schema to use. This process, using the adapter pattern, calls a SEXTANTE process which is creating the vector data. The output is a FeaturCollection in GML format.

ElevationZones Process

This process calculates and generates the elevation zones in vector data from a DEM file. It is a complex process like we define in the design section, internally this process in a chain of other processes. It calls the reclassify process and the vectorize process. As input parameters we have the URI to a DEM file in Geotiff format. The second input are the categories (elevation ranges), and since it will call the vectorize process, we need as input parameters the SRS and the application schema to be sent to the vectorize process which will generate the correct GML output parameter.

SRMIDLWPS

The SRMIDLWPS service contains a process to invoke IDL-based routines within the context of the SRM model. The Interface Data Language IDL is a language that traditionally has been used to process mathematical and image computations and is widely use in the hydrologist community. Some of the processes that were involved in the hydrological models where developed in IDL. For performance reasons we decided to leave theses processes in IDL, but we wanted to expose them as distributed web services as well.

The SRMIDLWPS service indeed wraps an IDL object as a web processing service by using the Java-IDL bridge component. In this way, it is possible to run IDL routines and handle IDL objects from Java code in a remote, distributed manner. However, it is important to note that this process requires that the host server will have to install a valid IDL licence for running IDL routines.

View Chart Services

Information representation as plots t is a strong requirement of scientists in general. Together with the hydrologists team we analyzed and defined all the plot and chart creation requirements of the AWARE application. The resulting processes have been grouped in the ChartWPS service.

ChartWPS

The plot functionality available in Chart WPS has been developed by using JFreeChart[xxxix] that is an open source java library that offers plot creation capabilities.

---

i http://www.flickr.com/
ii http://www.opengeospatial.org
[iii] http://www.inspire-geoportal.eu/index.cfm/newsid/4204
[iv] http://en.wikipedia.org/wiki/Net-centric
v http://www.geongrid.org/
[vi] http://www.dta.cnr.it/content/view/2735/2735/lang,en/
[vii] http://www.opengeospatial.org/projects/initiatives/ows-4
[viii] http://mapserver.org/
[ix] http://www.geoserver.org
[x] http://www.deegree.org/
[xi] http://geoserver.org/display/GEOSDOC/RESTful+Configuration+API
[xii] http://pywps.wald.intevation.org/
[xiii] http://grass.itc.it
[xiv] http://wpsint.tigris.org/
[xv] http://www.springframework.org/
[xvi] http://www.eu-degree.eu/DEGREE/wpx
[xvii] http://www.52north.org
[xviii] http://geocommons.com/
[xix] http://opengeo.org/products/suite/

[xx] http://www.esri.com/software/arcview/extensions/spatialanalyst/
[xxi] http://www.esri.com/software/arcgis/index.html
[xxii] http://www.autodesk.com/products
[xxiii] http://www.ittvis.com/idl
[xxiv] http://effis.jrc.ec.europa.eu/
[xxv] http://cwfis.cfs.nrcan.gc.ca/
[xxvi] http://afis.meraka.org.za/afis/
[xxvii] http://www.geomac.gov/
[xxviii] http://maps.geog.umd.edu/firms/
[xxix] http://javafx.com/
[xxx] http://openlayers.org/
[xxxi] http://www.gvsig.gva.es/
[xxxii] http://udig.refractions.net/
[xxxiii] http://geonetwork-opensource.org
[xxxiv] http://efdac.jrc.ec.europa.eu (last accessed on March 12th 2010).
[xxxv] http://en.wikipedia.org/wiki/Java_Native_Interface
[xxxvi] www.erdas.com
[xxxvii] http://www.opengeospatial.org/standards/sld
[xxxviii] http://www.remotesensing.org/geotiff/geotiff.html
[xxxix] http://www.jfree.org/jfreechart/