D.2.6.3. GEOSS Service Factory: Assisted content publication in GEOSS

Title: D.2.6.3. GEOSS Service Factory: Assisted content publication in GEOSS
Creator: L. Díaz
Creation date: 12/03/2012
Date of last revision: 13/03/2013
Subject: Publication, integration, hybrid building methodologies, participative GII, IOC, AOC
Status: Draft, Final
Publisher: EuroGEOSS, WP2
Type: Text
Description: Contributor: S. Trilles
Format: Doc
Source: Rights: Restricted, Public
Identifier: EuroGEOSS_D_2_6_3.doc
Language: En
Relation: WP2
Coverage: Not applicable

These are Dublin Core metadata elements. See for more details and examples http://www.dublincore.org/
TABLE OF CONTENT

1. INTRODUCTION ........................................................................................................... 3
2. SCOPE .......................................................................................................................... 3
3. MOTIVATION ................................................................................................................ 3
4. EXTENDING GEOSS SYSTEMS WITH PUBLICATION CAPACITIES .......................... 4
5. GEOSS SERVICE FACTORY DESIGN ........................................................................ 5
6. GEOSS SERVICE FACTORY COMPONENTS ............................................................. 8
7. USE CASE: PUBLISHING CONTENT ON FORESTRY INFORMATION SYSTEM ............. 10
8. DISCUSSION AND CONCLUSIONS ......................................................................... 12
REFERENCES ................................................................................................................. 13
1. INTRODUCTION

Information systems undergo a change from isolated solutions to open infrastructures based on Web Services. Geospatial applications have followed this trend for more than ten years to deal with data representing the status of our environment. International organizations and initiatives promote standards for data encodings and service interfaces that allow establishing Geospatial Information Infrastructures (GIIs). These GIIs provide services to address most of the steps in the geospatial user workflow, such as discovery, access, visualization and processing. However, they do not provide services to assist users in the publication of content. The lack of this functionality challenges the implementation and maintenance of GIIs since publication of content remains a complex task turning GIIs into top-down infrastructures without user participation. In this document we suggest extending classical GII architectures with a service that assists in content publication.

2. SCOPE

This document describes the research tasks carried out within the task 2.6 included in WP2 of the EuroGEOSS project. We address the publication of content to be integrated into GII and available in the EuroGEOSS project. Our research extends the EuroGEOSS Advanced Operating Capacity serving to a bigger purpose of providing multidisciplinary publication to increase availability of interoperable content in the EuroGEOSS systems.

3. MOTIVATION

Numerous geospatial content sources have to be managed to address challenges related to environmental monitoring. Global events, such as forest fires, impact different administration levels and it is important to identify risks and to provide early warning systems at these different geographic scales (de Groot et al., 2006). The current trend is to deploy and organize this information in Geospatial Information Infrastructures (GIIs) also known as Spatial Data Infrastructures (SDIs) (Masser, 2005). To increase the efficiency and interoperability of GIIs many regional and global initiatives work in the establishment of open standards and agreements. A GII on European scale has been legally mandated: the Infrastructure for Spatial Information in Europe (INSPIRE). It should provide environmental data related to 34 themes, including transport networks, land cover and hydrography, INSPIRE provides important parts of the European contribution to a Global Earth Observation System of Systems (GEOSS).

Initiatives, such as the two mentioned above, describe the overall architecture and best practices to design and implement GIIs. Content is managed by means of regulated and standardized service types. This imposes a distinct life cycle of geospatial content in distributed environments, which can be described in four steps as illustrated in Figure 1. First, content must be made available to a distributed system, i.e., content must be published in standard services like discovery and access services. Second, users need to discover content which will be finally accessed by using these services (third step). Finally, users process the content and generate new content, which should be integrated and published in the distributed system closing this cycle.
From our point of view, GIIs lack mechanisms to assist environmental experts, such as professionals in forest fire modeling, flood prediction or desertification, and casual users in the publication step. Such mechanisms should assist in making content persistent in a distributed and standard manner rather than storing content locally and isolating it from other users (Díaz et al, 2011). Thereby, GII users would become able to publish content to maintain GII up to date. Instead, the complex publication mechanisms of traditional GII, provoke a low-rate of user motivation regarding participation and content management (Coleman et al, 2009). We propose to extend GIIs to address these issues bottom-up. By doing so, we follow one main objective: Consideration of user participation to complement the traditional top-down implementations of GIIs by assisting users with publication mechanisms to improve content provision and therefore its availability. Hence, we introduce the GEOSS Service Factory (GSF). GSF assists users in the publication of content in existing standard service instances. It prepares a channel that hides complexity and facilitates content sharing, while remaining loyal to the geospatial initiatives agreements and standards to reach the required level of interoperability. This work extends and generalizes over our previous activities on a GII Service Framework (Díaz et al, 2011).

The GSF is our proposal to develop a generic publication service to assist users, both, experts of environmental domains or casual users, in content publication on certain system. To illustrate with a practical example we show the development and deployment of the GSF in a forestry fires system, where users benefit from this new channel to provide information. Compared to existing Geographic Information System (GIS) tools, which publishing geospatial data and maps using proprietary software, the proposed solution is implemented as a standard interfaced web service and can be generally applied a wide range of content types, service standards and information systems.

4. EXTENDING GEOSS SYSTEMS WITH PUBLICATION CAPACITIES

In the same way that GII provide users with functionality to perform daily tasks, such as discovery, access or download, we propose to provide users with content publication functionality. This functionality will be provided as an additional web service type deployed in the GII (Figure 2).
Following the trend in providing users with data discovery, visualization, downloads and other functionality as services, the GSF extend GII with *Publication as a Service* (green deployed service).

GSF is a standard and scalable publication service. It acts as a mediator to facilitate the publication of new content in GII as standard services compliant with INSPIRE and therefore with GEOSS. The term ‘GEOSS’ describes its intention on comprehend all the content types and services considered by GEOSS since they are more numerous, diverse and flexible than INSPIRE. The term ‘service’ describes its nature since it is a service and its main function is to update and provide new service content. The term ‘factory’ defines its behavior, it is designed to be a unique entry point to publish and modify different types of content.

### 5. GEOSS SERVICE FACTORY DESIGN

GSF behavior is modeled with the Abstract Factory design pattern to make it more scalable, and to extend the content types to a broader range. Furthermore, it provides the publication capacity with a standard OGC Service interface, the OGC WPS specification.

The *Abstract Factory* design pattern from software engineering (Gamma et al, 1995) is defined as a creational pattern used to instantiate new entities. It encapsulates a group of individual factories that have a common theme. In our case, the GFS holds a group of factories providing operations to publish new entries in Geospatial Services.

Figure 3 shows the Abstract Factory diagram adapted to our scenario. We use inheritance to derive the most specific OGC standards from the more general service types defined by INSPIRE and GEOSS. View and Download Services are some of the INSPIRE Service Types adopted in our approach to generate GEOSS Services and to deploy them in a certain GII. The individual services implement a particular OGC standard specification following INSPIRE guidelines to implement INSPIRE service types. For example, a WMS is used to implement a *View Service*, a WFS to implement a *Download Service*.
The GSF is implemented as a service component with a standard interface to be re-used in different scenarios. Since the OGC WPS is used to reach processing interoperability, it is our standard of choice to implement the GSF as a service. Figure 4 shows the UML class diagram with a simplification of the GSF interface and the signature of the publish process regarding input and output parameters.

We offer a single process, called Publish. The Publish process considers the following parameters:
Content: Only this input parameter is mandatory. This content can be passed by value or by reference, where a Uniform Resource Locator (URL) to the content can be used. It can vary from a vector or raster data set, or a metadata document.

ServicePublicationProfile: XML encoded parameter that describes the publication policy. This parameter includes information regarding where each data type should be published within this GII.

MD_URL: This parameter indicates that this content is already published in the GII and there are available metadata that should be reused when updating it.

Keywords: The optional 'keywords' parameter provides an initial capability for metadata creation.

DiscoveryLink: This is the only output parameter. This parameter contains the information needed to discover the content published in the system. In the case of the GII, where content is registered in Catalogue services, this parameter contains the end point to the metadata available in the Catalogue Service that contains the description of the content just being published.

Each application deployed in a GII can have its own publication policy. This policy establishes rules, for example, which content type is published in each service type that implement a particular specification where it is located. At technical level we describe this policy as a Service Publication Profile (SPP). SPP is set for a GSF deployed in a GII and it will configure the GSF to decide where each factory publishes the content. For example, the SPP determines which content types are published for visualization and download and whether it also publishes their metadata for discovery purposes. Next we can see an example of SPP:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ServicePublicationProfile
  <ServiceType name="Download" standard="WFS" version="1.0.0" url="http://elcano.dlsi.uji.es:8080/geoserver/">   <Property name="supportedFormat">text/xml</Property>   <Property name="supportedFormat">application/x-zipped-shp</Property>   <Property name="supportedFormat">application/zip</Property>   <Property name="supportedSchema">http://schemas.opengis.net/gml/3.0.0/base/feature.xsd</Property>   <Property name="supportedSchema">http://schemas.opengis.net/gml/3.0.1/base/feature.xsd</Property>   <Property name="supportedSchema">http://schemas.opengis.net/gml/3.1.0/base/feature.xsd</Property>   <Property name="supportedSchema">http://schemas.opengis.net/gml/3.1.1/base/feature.xsd</Property>   <Property name="supportedSchema">http://schemas.opengis.net/gml/3.2.1/base/feature.xsd</Property>   <Property name="supportedSchema">http://schemas.opengis.net/gml/2.0.0/feature.xsd</Property>   <Property name="supportedSchema">http://schemas.opengis.net/gml/2.1.1/feature.xsd</Property>   <Property name="supportedSchema">http://schemas.opengis.net/gml/2.1.2/feature.xsd</Property>   <Property name="supportedSchema">http://schemas.opengis.net/gml/2.1.2.1/feature.xsd</Property>   </ServiceType>
  <ServiceType name="Download" standard="SOS" url="http://urserver/sos">   <Property name="supportedSchema">http://schemas.opengis.net/om/1.0.0/om.xsd</Property>   <Property name="supportedFormat">text/xml</Property>   <Property name="supportedFormat">application/x-zipped-shp</Property>   <Property name="supportedFormat">application/zip</Property>   <Property name="supportedFormat">application/img</Property>   <Property name="supportedFormat">application/gdal</Property>   <Property name="supportedFormat">application/geoiff</Property>   <Property name="supportedFormat">application/dbase</Property>   <Property name="supportedFormat">application/remap</Property>   <Property name="supportedFormat">text/xml</Property>   <Property name="supportedFormat">application/geoiff</Property>   <Property name="supportedFormat">application/zip</Property>   <Property name="supportedFormat">application/dbase</Property>   <Property name="supportedFormat">application/remap</Property>   <Property name="transactionalProtocol">Geoserver</Property>   <Property name="password">pwd</Property>   <Property name="user">user</Property>   </ServiceType>
  <ServiceType name="View" standard="WMS" url="http://urserver/geoserver/">   <Property name="supportedFormat">application/x-zipped-shp</Property>   <Property name="supportedFormat">application/zip</Property>   <Property name="supportedFormat">application/img</Property>   <Property name="supportedFormat">application/gdal</Property>   <Property name="supportedFormat">application/geoiff</Property>   <Property name="supportedFormat">application/dbase</Property>   <Property name="supportedFormat">application/remap</Property>   <Property name="transactionalProtocol">Geoserver</Property>   <Property name="password">pwd</Property>   <Property name="user">user</Property>   </ServiceType>
  <ServiceType name="Discovery" standard="CSW" url="http://urserver:8080/geonetwork/srv/en/csww">
```
6. GEOSS SERVICE FACTORY COMPONENTS

The basic idea is that the GSF will publish content creating new entries in existing Services. We have model its behavior with the design pattern Abstract factory to make it more scalable, therefore it is composed by different factories each of them dealing with the publication of content in a different service type. Besides the Factories the GSF contains two other modules, the transformation module which deals with data format transformation to overcome the problem of spatial formats not being supported by existing services. The MD generator is able to generate a small discovery purpose metadata of the published content; this metadata can be automatically published to Discovery Services via the DiscoveryFactory.

As shown in Figure 5, when the GSF is invoked, the content to be published is sent as an input parameter. The GSF settings are stored in the PublicationProfile, which reflects the publication policy of the GII. It says in which service type is going to be published this content which is the service interface and where is the instance running.

Then the GSF delegates the content to the corresponding factory depending on the content type. The factory will communicate with the existing services via the services APIs, most of them implement OGC interfaces, but other standards or service specifications can be added.

After the content is published, the GSF can delegate on the discovery factory to register it as we can see in Figure 6. In this case the DiscoveryFactory through the MDGenerator queries the updated services to extract discovery metadata information. The metadata will be published in a Catalogue for further discovery as we describe in the metadata generation section. The output of the publish process can be a set of links or a link to the metadata entry containing the links to the content and some description.

To generate the metadata information the data and the site where it is published must be specified. Part of this information can be extracted from the services where the data has been published. These data services are mainly available in SDIs.

These services conform to the INSPIRE implementing standards are as OGC services WMS, WFS, Web Coverage Service (WCS) and SOS. Other information that is more difficult to identify
are the keywords, so we designed GSF with input parameters such as keywords, which may be indicated by the user.

In general, the metadata generator will have to run when the data is published, if not stated otherwise. The factory responsible for working with metadata generator is the Discovery Factory.

The first step is to consult WMS and WFS services to get the information needed to create the metadata. This is done through appropriate requests (GetCapabilities). After obtaining the necessary information the process applies metadata generation process and proceeds to their publication in the catalog (CSW). Finally, the WPS will return the URL result to the publication. This operation can be seen in Figure 6.

![Figure 6: Steps for the publication and generation metadata in the GSF.](image)

After publication of content, if the WPS has not received a MD_URL parameter, metadata is then called by the metadata generator. This requires the URLs to public services to generate the requests, in addition to input keywords. The only output parameter of the GSF has a URL to access the metadata catalog service where it has been published. To further explain the process of metadata generation, we have the following image (Figure 7a) which shows the inputs and outputs of the Metadata Generator module. Figure 7b shows the steps involved.

![Figure 7: a) Input and output parameters metadata generator b) Steps metadata generator.](image)

In the first step the GetCapabilities request is performed to services of information where the previous content has been published. For example, if the content has a vector, this will be
published for visualization in WMS and for download in WFS. This request returns a XML with the characteristics of the information that will be necessary for the metadata creation. Secondly, the same step is done by the WFS, to provide more information to the metadata.

The following step must apply the Extensible Stylesheet Language Transformations (XSLT) to generate metadata. The transformation will take the XML entry obtained from the previous request. The result obtained after the transformation is another document XML, which is the metadata with the ISO 19139 standards and INSPIRE.

The third step consists of parsing the XML metadata for fields that have not been filled in the transformation, such as keywords or URLs to services. After this step, the metadata will be considered completed.

Then the metadata will be published in the catalogue that is defined in the SPP. This step is performed similarly to the publication of data but in this case to the catalog. In our work we use transactional profile (CSW-T) according to the OpenGIS Catalogue Services specification. In this way, the system will have to implement all operations allowing the CSW-T protocol. These are: GetCapabilities, GetRecords, GetRecordById, GetDomain, DescribeRecord and the transactional.

The last step is to get the URL that identifies the metadata posted on the server. After obtaining the metadata URL, the generator metadata is considered as finished.

For the generation of metadata has been chosen to create the template using XSLT. This technology is a standard from the WorldWideWeb Consortium (W3C) organization that allows us a way of transforming XML documents or other types of documents. As we have already stated metadata is nothing more than an XML following a standard so that we can apply this type of transformations.

We should mention that the metadata that is generated does not have all the fields in the specification and some cannot be obtained automatically. We have chosen the fields you can see in the figure below (Figure 8). It shows the main fields that contain the metadata generated.

![Figure 8: Some metadata fields.](image)

7. **USE CASE: PUBLISHING CONTENT ON FORESTRY INFORMATION SYSTEM**

Fire information systems exist from local to global scale, for example EFFIS (European Forest Fire Information System) at European level and FIRMS (Fire Information for Resource
Management System) at the global scale. A common challenge for these systems is to provide geospatial data in a quick and reliable way before and during fire emergencies. It has been demonstrated that geospatial support systems implemented with interoperability standards provide easier access to this type of critical information (Friis-Christensen et al., 2006).

All of these systems offer functionality to access forestry and fire resources in a standard basis. In EFFIS, for example, information layers (raw and processed data) are presented into the map viewer through an internal WMS which will be publicly available among other standard services, such as WFS (Giovando et al. 2010).

Among other functionality, such as data visualization and download, EFFIS incorporates internal processing, such as burned areas calculation and the forecast of fire danger. Scientific users generate and modify content, but are IT specialists who have to publish and update this content. As mentioned before this impedes users’ publication and participation and provokes a bottleneck in the publication process. Therefore, we propose to extend EFFIS with GSF as a publication service.

Figure 9 shows the existing EFFIS components based on the Forestry Initial Operating Capacity (FIOC) (EuroGEOSS, 2010).

Existing components focus on the content being available through standard services for searching and visualization. We extend the system with a publication service as it is shown in the top of the figure. GSF is deployed in the system as any other service. As the ‘Map Server’ and the ‘Metadata Server’ are associated to their content sources, the GSF is associated to its content sources; these are the (OGC) services in which GSF will publish the content. As it is shown in the figure, the publication profile of the EFFIS indicates that the GSF will publish content in to the Map Server and the Metadata Server.

![Figure 3: Example of how to extend an existing system with a Publication Service: GEOSS Service Factory integrated in EFFIS.](image)

At the top layer, GSF has to be integrated into client applications. As we implement the GSF as a WPS, functionality can be accessed through a WPS client. We have implemented such a client as a Java library that is part of the Service Connector (see Figure 2), to be re-used in any Geoportal. Figure 9, shows an overview of EFFIS extended with current functionality of the GSF prototype.
The figure shows how scientific users can generate new content, for instance performing processing (local or distributed), such as the calculation of burned areas, by accessing existing EFFIS services. GSF provides a new channel for assisted publication of certain data types, such as a burned area map. In this way, GSF offers a unique entry point with a standard interface to publish content in the EFFIS system according to a publication policy.

We have set the Service Publication Profile of the GSF for the forest fires system. View, Download and Discovery factories are associated with a View, Download and Discovery service instance deployed in EFFIS. When users generate new content, for instance, by running an environmental model to calculate burned areas, the extended EFFIS provides publication functionality, the system top layer (Geoportal) can connect the user to the GSF (by means of the user interface and the WPS client API as shown in Figure 9). The GSF receives the ‘Content’ parameter and publishes the dataset in a View, Download and Discovery Service using the individual factories that connect to OGC WMS, WFS and CS-W instances respectively. The GSF output is a URL to the newly generated metadata element that is available at the CS-W. This element contains information of how to view and download the dataset published in the OGC services.

For the prototype, the View Service Factory and Download Service Factory publish WMSs and WFSs based on GeoServer technology\(^1\), specifically these factories publish content using the GeoServer RESTful API to connect and update the different service instances. In order to deploy new metadata, the Discovery Service Factory implements the transactional interface of the CS-W being independent of the CS-W implementation instance used in EFFIS. For testing purposes we have used Geonetwork\(^2\) as CS-W implementation.

8. **DISCUSSION AND CONCLUSIONS**

This demonstrator has shown the research tasks and the components developed within the task 2.6 in WP2 of the EuroGEOSS project. This document has presented a prototype that illustrates the current work. We have presented an approach and developed a mechanism to assist in the publication of content in GEOSS and INSPIRE based information systems.

We argued that content publication to GIIs causes the central bottleneck in environmental information sharing. Bottom-up approaches are needed in the context of GIIs to assist users in populating these infrastructures with content. Following the work of international activities, the GEOSS Service Factory (GSF) was promoted as a solution. We proposed the GSF as a publication service of GII architectures. In order to ensure platform independence, it is provided as a separate component that becomes accessible through standard interface such as OGC WPS from the Geoportal front-end of a GII. In future a possible specialization of this interface can be developed using profiling, to offer a common profile for content publication in GII. In this work a core operation called ‘Publish’ and its mandatory and optional parameters have been defined. We recommended the deployment of INSPIRE Service Types, because INSPIRE provides a formal framework to GEOSS while at the same time providing an abstraction layer on top of OGC standards. A prototype has been depicted as proof of concept.

Our proposal alters the role of GII users, being either professionals or casual users. They turn from rather passive consumers into active participants playing a more interactive role and providing new content (Budhathoki et al, 2008). Now, users can participate in the maintenance and updating of the GII. This means that users, besides searching, accessing and analyzing data, could massively publish newly generated content as interoperable components. This would improve the availability of interoperable content in global, regional and local services related to domain specific scenarios.

---

1. [http://geoserver.org/display/GEOSDOC/RESTful+Configuration+API](http://geoserver.org/display/GEOSDOC/RESTful+Configuration+API)
2. [http://geonetwork-opensource.org](http://geonetwork-opensource.org)
and could increase the effectiveness of GIIs. The monitoring and reporting of forest fires in EFFIS provides one example.

The approach presented raises collateral issues, such as security and quality assurance, which need to be put in place in order to assure that the integrity of GII content. This document already provides a brief outline of how the publication policy in a system could be kept by configuring the GSF with authentication parameters. Furthermore, GSF is an extensible component, which could be extended with modules for content validation.

The suggested flexible implementation of the GSF offers possibilities for resource plug-and-play. New factories can be added using class inheritance. This may be extended in order to publish other content types, such as environmental simulations and VGI, including geospatial data models extended with uncertainty information (Williams et al. 2008) or non-geospatial content, into GIIs. The suggested approach also provides flexibility in terms of functional extensions, such as content validation, security, and automated reasoning capabilities.

Regarding metadata generation we have described an automated creation of metadata (to be send to the geospatial catalogue). First implementations show that it is feasible to extract essential information from some geospatial data encodings, but more detailed elaborations are still required.

With the approach presented, we move closer to real usage of GIIs, because end users become involved in content provision. Once the barrier of motivation has been overcome, we will be able to benefit from GIIs for effective and efficient information sharing, one of the main goals of GEOSS.

REFERENCES


forest fire assessment example. 9th AGILE International Conference, Visegrad, Hungary, pp. 119–27.


