WMS Integrator: continuous access to neighboring WMS

Miguel-A. Manso-Callejo(1), María-T. Manrique-Sancho(1), Paloma Abad-Power(2)
(1) Dpto. Ing. Topográfica y Cartografía, UPM, Ctra de Valencia Km7 Madrid 28031, ETSI en Topografía, Geodesia y Cartografía, m.manso@upm.es, maytemanrique@topografia.upm.es
(2) Dirección general del Instituto Geográfico Nacional, C/ General Ibáñez de Ibero, 3 28003, Madrid
pabad@fomento.es

ABSTRACT

As a consequence of consensus and experimentation processes, OGC has defined several versions of the WMS. The INSPIRE Directive is working on a set of implementation rules designed to harmonise the types of data and to standardise the services providing them, among other purposes. In order for users to be able to continuously (concurrently) exploit WMS offered at the different SDI hierarchical levels, a *facilitator service* is needed making the use of multiple service URLs, layers, CRS, etc. transparent. This *facilitator service* should integrate the cascade responses to a series of remote WMS in the form of a layer group, exploiting the possibility foreseen by the standard. This document presents the research work carried out to develop a facilitator WMS integrating horizontally and vertically the layers of lower geographic hierarchy provided by the WMS, obtaining as a result one single seamless layer. One of the most salient contributions is the capability of restricting the response of a remote WMS in the merging process by means of a polygon, thus avoiding the overlap of information or the lack of transparency of the responses.

Keywords: WMS, Integration, Facilitator, Cascade WMS, Administrative boundaries

INTRODUCTION

The first two tenets of the INSPIRE Directive (Directive 2007/2/EC) [1] are: 1. Data should be collected once and maintained at the level where this can be done most effectively; 2. It should be possible to combine seamlessly spatial data from different sources and share it between many users and applications)[2]. The visualization or the visual exploitation of Web geographic data has been standardised by OGC as WMS (Beaujardière, 2006), by ISO 19128 as an international standard and finally it is being adopted by INSPIRE as view services.

When users/applications need to see/handle continuously the GI of a particular topic, whether included or not in the datasets Annex I-III of INSPIRE directive, first thing they have to do is to search in the discovery service for the access points to the WMS that enable its visualization. After the services have been identified, the layers containing the desired data should be selected, and then as many *GetMap* type requests are triggered as services are affected by the geographic context object of the request. At this point a set of difficulties may be enumerated indicating the need for the development of a *facilitator service* that should carry out some verification, requests to the different involved visualization services and merge the responses to finally deliver a single image as a result.

The main identified difficulties are the different versions supported by the WMS Specification (1.1.0, 1.1.1, 1.3.0) that imply changes in the syntax of some parameters in the requests, the CRS provided by the different services that may involve coordinate conversions/transformations on the images obtained from the services, the graphic formats offered by the services (GIF/JPEG/PNG), some of

[1] http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2007:108:0001:0014:EN:PDF
[2] http://inspire.jrc.ec.europa.eu/index.cfm/pageid/48

which do not support transparency, and the response times acceptable for the services. In our case, we also want to avoid overlapping geographic information provided by the WMS of the adjacent regions.

This document presents: a) the results of the review made on this issue and the capabilities and limitations of WMS for the management of remote layers in free software implementations; b) the design proposal of a WMS solving the problem of continuity in the access to data offered by local, regional or national WMS; c) the service developed on the basis of the design and, d) conclusions and possible applications of the service developed.

RELATED WORK

One of the first found references to WMS integration dates back to the Memon brothers (2005) in discussion documents related to cyberinfrastructure for the GeoSciences (GEON: GEOscience Network)[3]. The objective is to develop an infrastructure of distributed GI with GRID technologies. In this work an architecture is presented in which, as a consequence of a map service search in the catalogue WMS URL, service envelope, supported WMS version, layer names, layer styles and layer projections are obtained. These data are supplied to the WMS Integration Service for the realization of GetMap requests to the different WMS and composition of the resulting image. The authors pinpoint in this service to a number of problems that have to be resolved: projection negotiation, layer sequencing, quality of service and granularity of integration. After identification of the conflictive aspects, the problems coming up as a consequence of the development of an intermediate solution are analysed: lack of compliance with WMS standards, lack of a flexible programmer interface, lack of scalability, no incorporation of the notion of WMS layers, no support for legend or for identification of the operation even if the layer does support it. At this point they propose the use of the OGC Standards as WMS clients and the use of the Web Map Context (WMC) (Sonnet, 2005) as output of the WMS integration service, i.e. they displace the complexity of the integration to the client side that should be conformant with WMC and, in addition to being flexible to manage layers/groups of layers, it should allow identification queries directly to the affected service. The projection changes, when none of the coordinate systems offered by a service to be integrated is the one making the request, remain out of the scope of the mentioned proposal, as well as the possibility of defining geographic limits for the responses offered by the integrated services.

The second work found in the same line is the one started by Zevenbergen *et al.* (2007). They state: *"A more difficult issue is the integration of the datasets covering the same topic but different areas. A WMS-integrator was developed to overcome the technical issues of integration but more importantly, what still remained is the integration of the content."* The WMS Integrator (WIS) [4], such as defined, may carry out horizontal and vertical integration of layers. The horizontal integration consists of adding a layer of each integrated service on the same layer of the final service. The vertical integration consists of joining several layers of the same service as if they were only one. Figure 1 illustrates these concepts.

The proposed solution in GeoLoketten implements in a WMS service (v. 1.1.1) the integration of the maps generated by the involved WMS. Unlike the Memon's proposal (2005), GeoLoketten proposes to generate a new service accomplishing the integration and responding in a standardised manner. The lack of information about WIS does not allow knowing the type of technology used or how to configure the service. Taking into account those considerations, our hypothesis about its implementation is that the integrator service does not consider the possibility of the involved WMSs providing GI outside of the geographic domains it manages.

[3] http://www.geongrid.org/
[4] http://www.geoloketten.nl/wms_integrator_service.html

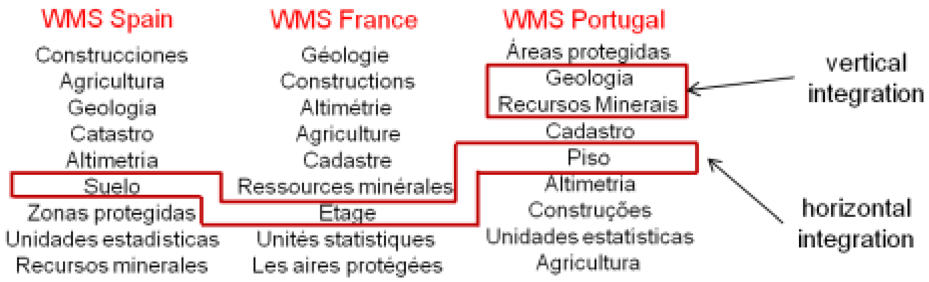

Figure 1: Map service: Horizontal and vertical integration (based on GeoLoketten)

Another analysed document is "Aggregating Remote Map Services with Local Cascading WMS Server" where Brunclik (2007) describes the functions for which the Cascade functionality, anticipated by the WMS Specification, is commonly used. In that document mention is made of the services that are utilized to make more versatile the results of WMS not supporting certain output formats, new service versions or not being able to transform data between reference/coordinate systems. They are also used to build a wrapper making a service standard when it was not.

After having identified the possibility of providing layers of other WMSs as remote layers or in cascade, the capabilities offered by the most popular free software projects to serve or handle those layers has been studied, namely MapServer[5], GeoServer[6], degree[7] and SimpleMapServer[8]. It has been concluded that only MapServer and degree may offer them. In GeoServer a query thread has been located in relation to that capability, although there are no examples or documentation describing how to go about it. After having analysed the documentation that describes how to carry out the configuration, some experiments have been made to integrate horizontally a set of topographic cartography layers offered by two neighbouring autonomous communities, with scales 1:5,000 and 1:10,000. The conclusions of these experiments are:

- MapServer may provide layers coming from other WMSs identifying service URL, layers, SRS, formats, transparency, connection waiting time, service version and geographic extent of the layers through a BBOX.
- Deegree may provide layers coming from other WMSs identifying service URL, layers, SRS, formats, transparency, connection waiting time, service version and area in which the cascade service is valid through a polygon described by GML (*ValidArea*).

According to this observation it may be stated that deegree is more effective by more accurately limiting the geographic extent. This enables to better evaluate if the client's requests intercept or contain *ValidArea* data. However, deegree does not use that polygon to mask information not contained within. A disadvantage of deegree lies in the initialisation process of the WMS service; in this process a GetCapabilities request is made to the remote nodes and if the resulting document is not well shaped or the links to its operations (DCPType:: OnlineResource) are not well defined, the WMS-Integrator service will not be correctly initialised.

[5] http://mapserver.org/
[6] http://geoserver.org/
[7]http://www.deegree.org/
[8] http://glmapserver.sourceforge.net/

In the cases of MapServer and deegree it has been confirmed that the overlap of the GI provided by the underlying WMSs persists as a consequence of providing data outside of its area of responsibility (e.g. the boundaries of the autonomous communities). Figures 2 and 3 show this GI overlap.

Figure 2: WMSs of the Autonomous Community of Euskadi (a), Navarre (b) and Overlap of both (c)

Figure 3: Boundaries of Autonomous Communities (a), WMS of Aragón (b), La Rioja (c), Navarre (d) and overlap of them (e)

PROPOSED DESIGN FOR THE WMS-INTEGRATOR

Before presenting the design of the WMS-Integrator, the main requirements and objectives to be achieved, and later its implementation are exposed:

- The service must offer an access interface based on the WMS Specification of OGC (Beaujardière, 2006), admit negotiation of versions (1.1.0, 1.1.1 y 1.3.0) and at least implement the mandatory service operations (GetCapabilities and GetMap).
- The service will offer one single layer as a result of horizontal and vertical integration (within a same service) of the integrated WMSs. The layer name will be defined in the configuration file. The polygons defining the geographic boundaries of the remote WMSs will be stored in a shapefile. The access path and the filename will also be defined in the configuration. For every polygon describing the remote WMS, all the necessary information will be stored: URL, layers, supported SRS, image format, service version, maximum waiting time, etc.
- The metadata describing the integrator service and their capabilities will be defined in the configuration file.
- The language of development will be Java.

Figure 4 shows the high level architecture of the Web service that performs the integrating function. The two modules making up the service are identified: the integrator module and the MapScript module with the MapServer libraries. The MapScrip module will be charged with the management of the requests to the remote WMSs and it will draw the masks defining the boundaries of the geographic domains of every remote WMS. The integrator module will implement the interface proposed by the WMS Specification and will take care of the requests. This module should intercept the requests (GetMap, GetCapabilities, etc.), identify the polygons affected by the request, generate as many requests to the remote WMSs and masks as remote services are affected, and finally group together all the responses after masks have been applied on one single image; that will be the response of the integrator service. Figure 5 shows a diagram of activities and Figure 6 shows a diagram of sequences.

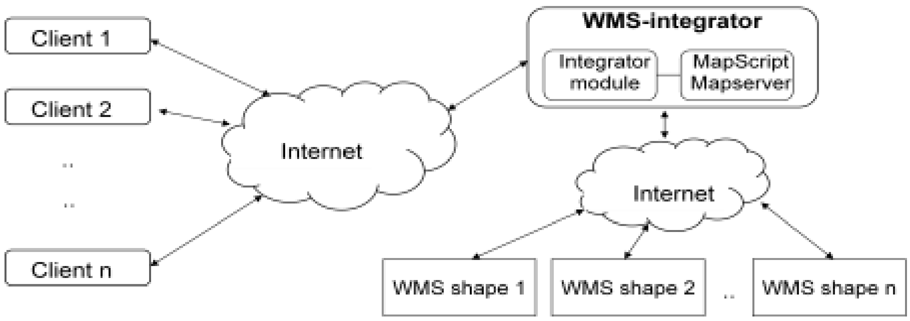

Figure 4: High level architecture of the integrator service

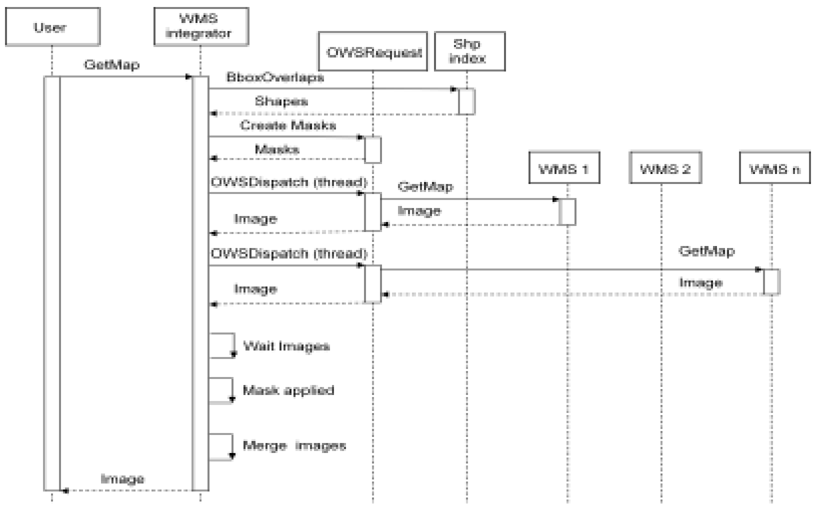

Figure 5: Diagram of activities

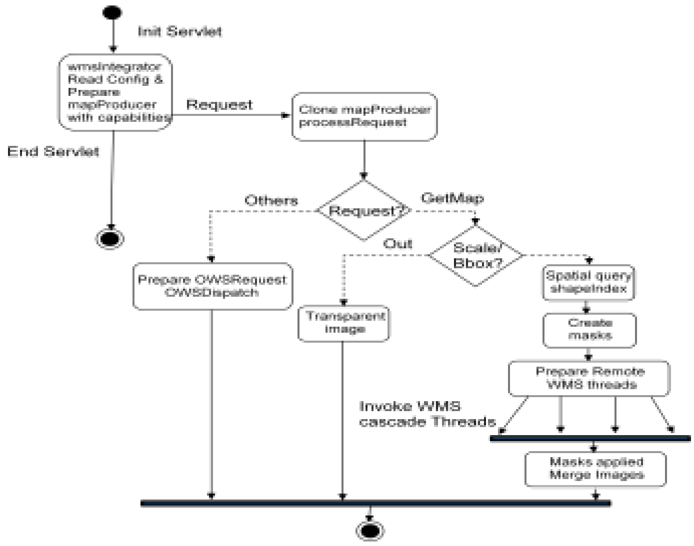

Figure 6: Diagram of sequences

After the diagrams of activities and sequences have been designed, the application diagrams of classes are presented. First the diagram of classes for the information to be stored in the shapefile identifying the cascade WMS services is shown. As can be observed in Figure 7, the attributes associated to every polygon are: *wmsUrl*, *wmsVersion*, *layers*, *styles*, *srs*, *format*, envelope {*minx miny maxx maxy*}, *reqTimeout* and *wmsTitle*. We are dealing with information required to dynamically build a request to the remote WMS service and to respond to the GetCapabilities requests.

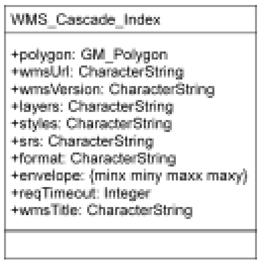

Figure 7: Diagram of classes of the shapefile that indexes the remote WMS

In order to meet the requirements relative to configuration and metadata of the integrator service, the following classes have been defined: Service_configuration, Service_logging (shown in Figure 8), WMS_Name, Service_Metadata and Layer_Metadata (shown in Figure 9).

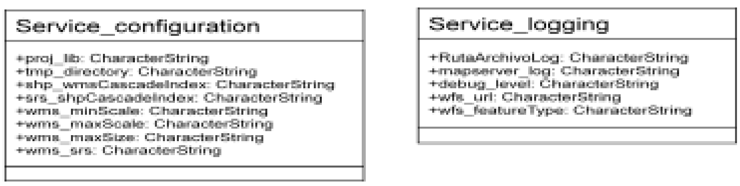

Figure 8: Diagram of classes for Service_configuration and Service_logging

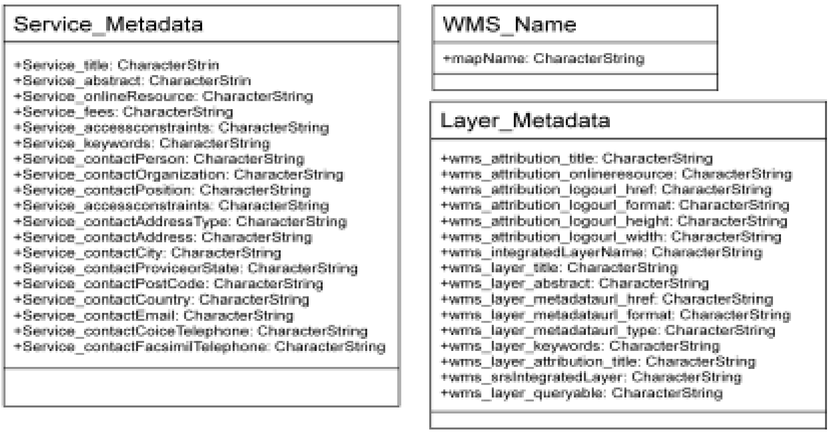

Figure 9: Diagram of classes for WMS_Name, Service_Metadata and Layer_Metadata

DEVELOPED FACILITATOR WMS SERVICE

In this section the characteristics of the prototype service developed are described based on the MapScript[9] libraries (Java) of the MapServer Project (v5.4). The first step of the development has been the compilation of the library using the VC++6.0 compilator for Windows and the compiled libraries of MapServer, GDAL and OGR emanating from the FwTools[10] package. The development environment used is NetBeans[11] and the environment of execution for the servlet developed is Apache Tomcat[12] (v6.0).

The created servlet (wmsIntegrator) reads from its initial directory a configuration file (wmsIntegrator.properties) in which the information described in the classes (Service_configuration, Service_logging, WMS_Name, Service_Metadata and Layer_Metadata) is stored by means of keywords. When the servlet starts, all the information is read and processed, so that when a request arrives at the service, the necessary information is in the memory and the only thing it does with it is to clone it. The arrival of requests causes the generation of a new object of the wmsIntegratorMapProducer, responsible for the coordination of the operation flow in its entirety. The type of request is discriminated in this new object and the request is responded accordingly. With GetMap type requests the first step is to validate the query and verify that the scale and the spatial context are correct. Next the affected remote WMSs are calculated spatially and threads are created and launched that retrieve their images. The process awaits arrival of all responses or the timeout established for each of the WMSs through a synchronisation mechanism. Once the responses are received, the masks are applied over each one of them and the results are merged. Finally the resulting image is returned to the client and the life time of the object ends.

In order to verify if the requests are correct, a Map type object is established with the parameters obtained from the configuration, an OWSRequest is built with the parameters provided by the client and the execution is requested (OWSDispatch). If the response is correct the calculations are carried on, otherwise the resulting exception message is returned.

---

[9] http://mapserver.org/mapscript/index.html
[10] http://fwtools.maptools.org/
[11] http://www.netbeans.org/
[12] http://tomcat.apache.org/

In order to find out if the request scale is correct, the maximum and minimum coordinates of the request are transformed/converted into geographic coordinates, the criterion defined by ISO 19128 (Section 7.2.4.6.9) for calculation of scale is applied and subsequently comparison is made with the scale range established in the configuration.

In order to find out if a request is out of the geographic context offered by the service, the maximum and minimum coordinates of the request and the maximum extent, defined in the configuration shapefile WMS_Cascade_index, are transformed to the same coordinate system. Then, a spatial query of intersection and content is made. In the event that the request is not correct because of inadequacy of the scale or its geographic definition, the implemented solution has been to return the image empty and transparent, so that the response of the service should not interfere with other responses in the client.

Since the integrator service has been developed on the basis of MapServer, other optional operations of the WMS service defined in v1.1.1[13] have been implemented, such as getSchemaExtension, describeLayer, getLegendGraphic and getStyles. In addition, a non-standardized operation (getStatus), enabling to query the service to know the status of the remote services has been implemented. The service makes a GetCapabilities request to everyone and interprets the response to detect whether the server is active or has undergone changes in its configuration making it incompatible with the current configuration of the integrator service: layer names, CRS, formats, etc. Figure 10 shows the diagram of interfaces of the developed service.

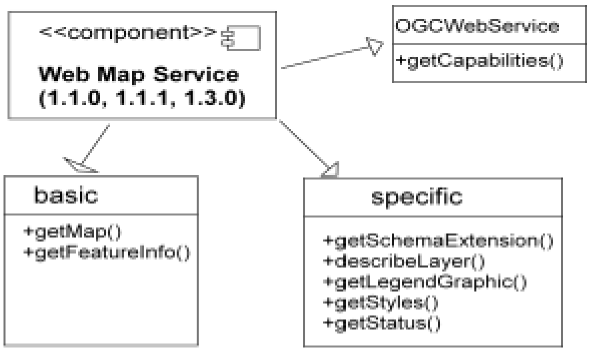

Figure 10: Diagram of the developed WMS service

The display of the WMS Integrator service presented here requires, in addition to the MapScript-based libraries used (mapscript.dll, mapscript.jar, libmap.dll, etc.), a configuration file (wmsIntegrator.properties) and the files associated to the shapefile format supporting the WMS_Cascade_index class. Two important elements of the configuration are the definition of a temporary directory to store the remote images needing rectification and the definition of the path to get to the Proj[14] libraries.

RESULTS AND CONCLUSIONS

The subject of WMS integration and the capabilities of free software projects to use remote layers (Cascade_WMS) have been reviewed. A new WMS has been designed and developed with capability

[13] http://portal.opengeospatial.org/files/?artifact_id=1081&version=1&format=pdf
[14] http://trac.osgeo.org/proj/

for horizontal and vertical integration of one single layer and the layers provided by a set of remote WMS covering the same topic but different areas.

Unlike other proposals, the developed service is capable of avoiding the information overlaps that the WMSs of different areas have outside of the geographic domain of their competence. Figures 11 and 12 show two graphic examples of the responses generated by the service in the conflictive cases represented in Figures 2 and 3 respectively.

Some conclusions and remarkable characteristics of the service are:

- It is based on a consolidated free software project (MapScript-MapServer), thus the service may be deployed as many times as desired without increasing the cost of licenses.
- The quality of the service in terms of response times is not good and is due to the long response times of the subrogated services. In our case the worst response time is the highest value of the timeouts established by the remote WMSs.
- To improve the benefits of the WMS Integrator, service replication may be considered leaning on the first conclusion and on the fact that the service *per se* does not provide data but it distributes work, adds value and delivers results, all of which facilitates replication in a GRID computing environment.
- The developed service may also be used to generate a WMS-Tiled service cache, so that the benefits of the service as a whole could be remarkably improved.
- The topic/content of data that this service integrates is defined in a shapefile in which every polygon identifies the geographic extent valid for integration. Associated to these polygons, the URL, layers, SRS, styles, etc. are stored. In another configuration file the service metadata are defined and some working values are established.
- Leaning on the previous characteristic, the developed service enables deployment of new thematic integration services by an application/intelligent service on the basis of the results obtained from queries to service catalogues and download services associated to the topic of administrative boundaries.
- Ultimately users, through WMS clients – thin or thick – may interact with the integrated services in a transparent, simple manner since the interaction is only with a service that implements all the logic of negotiation and the necessary transformations.

Figure 11: Integration of the WMS of the Autonomous Communities of Euskadi and Navarre

Figure 12: Integration of the Autonomous Communities of Aragón, La Rioja and Navarre

ACKNOWLEDGEMENTS

This work has been funded by the España Virtual CENIT Project, which was sponsored by the CDTI within the INGENIO 2010 Programme, through the CNIG.

BIBLIOGRAPHY

Beaujardière, J. (2006). OpenGIS® Web Map Server Implementation Specification. Editor.

Brunclik T. (2007). Aggregating Remote Map Services with Local Cascading WMS Server. Proceedings of the 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases, Corfu Island, Greece, February 16-19. Pg: 250-254

Memon, G. and Memon, S. (2005). WMS Map Integration – Improved. SDSC GEON Weekly Meeting Webcast, November 2005, La Jolla, CA

Sonnet, J. (2005). Web Map Context Documents. OGC 05-005.

Zevenbergen, J., Bulens, J., Kuyper, M., Koerten, H., Jellema, M. and Welle-Donker F. (2007). "GEOPORTAL NETWORK" – MORE PROCESS CATALYST THAN PROJECT