

Towards an ESB-Based Enterprise Integration Platform for Geospatial Web Services

Bruno Rienzi

InCo, Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
brienzi@fing.edu.uy

Laura González

InCo, Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
lauragon@fing.edu.uy

Raúl Ruggia

InCo, Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
ruggia@fing.edu.uy

Abstract— Geographic Information Systems (GIS) have emerged to store, handle, analyze, and present geographic data to experts and casual users alike. As the number and scope of geo-enabled applications have increased considerably in recent years, new software solutions, like Internet Map Servers (IMS), have been incorporated into the enterprise software portfolio, giving birth to the “Enterprise GIS”. Although these systems are based on standards for distributed geospatial processing and data exchange, through the use of Geospatial Web Services, they have to be complemented with other mechanisms to integrate them with business data and functionalities provided by traditional enterprise systems. In turn, an Enterprise Service Bus (ESB) is a standards-based integration platform, which provides mediation capabilities to address mismatches among applications regarding communication protocols and message formats, among others. This paper proposes an ESB-based reference platform which, leveraging its mediation capabilities, provides reusable geo-oriented integration mechanisms to deal with common challenges of integrating traditional enterprise systems with Geospatial Web Services.

Keywords-gis; enterprise applications; integration; geospatial web services; enterprise service bus.

I. INTRODUCTION

Traditionally, a Geographic Information System (GIS) has been defined as a system which integrates hardware, software, and data for capturing, managing, analyzing and displaying all forms of geographically referenced information (for a more extensive definition, see [1]). Looking back just a couple of decades, GIS was mainly provided by desktop applications, which stored data in files using proprietary formats. GIS was clearly geared towards experts. In the past decade, however, this situation has changed. It has been increasingly understood that location, as well as time, is a ubiquitous dimension of almost all data. As a result, many enterprise technologies, like relational databases and application servers have adopted GIS components to provide Geographic Information (GI) to a broader number of users. Enterprise GIS was born [2].

Like any other application within an organization, a GIS needs to be integrated with other software systems. In order to facilitate this integration, the Open Geospatial Consortium (OGC) has defined an architecture for distributed geospatial processing and data exchange, based on its own set of Web

Services (WS) which are known as OWS (OGC Web Services).

In a simple scenario, an organization has its geographically-related data tied together in the same data source, typically a spatial database, and deploys off-the-shelf products, such as an Internet Map Server (IMS) that will query the spatial database to publish the GI, and a Map Viewer that will communicate with the IMS through OWS to allow user interaction. However, it is often the case that a map has to show additional information that comes from several data sources. For instance, a certain company could be interested in using an addresses map (or, more precisely, an addresses *layer*) that contains all the geo-coded addresses in a country and link those addresses to its clients data to perform certain analysis on the map. As the clients' data may spread over the company's CRP databases and external systems (only accessible via WS), an IMS alone cannot cope with this integration requirement.

While several approaches [3] can be followed to integrate heterogeneous data and services in a complex scenario involving geographic and non-geographic information systems, the absence of standardized mechanisms leads to high costs and complexity in developing home-made integration solutions, which suffer from strong limitations in terms of service-oriented connectivity and reusability.

In this work, an Enterprise Service Bus (ESB) is used as the basis of an integration platform that addresses the aforementioned limitations in current systems. An ESB is a standards-based integration platform that combines messaging, WS, data transformation, and intelligent routing to reliably connect and coordinate the interaction of diverse applications [4]. An ESB provides a middle integration layer, with reusable integration and communication logic, which helps to address mismatches among applications regarding communication protocols, message formats, and quality of service (QoS), among others [5].

This paper proposes an ESB-based platform which extends the basic mediation capabilities with reusable geo-oriented integration mechanisms to deal with common challenges of integrating traditional enterprise systems with OWS, focusing on providing a reference architecture for this platform and describing various concrete geo-oriented integration mechanisms.

The rest of the paper is organized as follows. Section II provides background concepts. Section III presents a solution approach to address the aforementioned integration

challenges. Section IV proposes and describes various concrete geo-oriented integration mechanisms. Section V presents implementation details. Finally, Section VI presents conclusions and future work.

II. BACKGROUND

This section provides background on OWS and ESB mediation patterns. These technologies constitute the basis to build the proposed solution.

A. OGC Geospatial Web Services

OWS [6] have coarse-grained interfaces with a few stateless operations. They use HTTP and XML, but not SOAP or WSDL (the W3C standards for WS).

This paper focuses on two OWS standards: Web Map Service (WMS) and Web Feature Service (WFS). WMS [7] produces maps dynamically. A map, in WMS terms, is a portrayal of GI as a digital image file, which is the result of overlaying several geographic layers. A layer is collection of features of the same type (e.g., a roads layer). A feature [8] is an abstraction of a real world phenomenon (e.g., a road). WMS provides two specific operations: the mandatory GetMap and the optional GetFeatureInfo. A GetMap request specifies, among others, the layers to make up the map, the coordinate reference system (e.g., Universal Transverse Mercator) and the geographic area (e.g., the rectangular boundaries of a country). A GetMap response is the map itself. On the other side, a GetFeatureInfo request specifies the coordinates of a point and a GetFeatureInfo response brings the attribute data of the features that contain or are near that point (as long as the features belong to a *queryable* layer). The attribute data of a road could be its name, type, average traffic, etc.

WFS [9] offers direct fine-grained access to GI at the feature and feature property (attribute data) levels. It allows clients to retrieve, create, modify and delete features, using XML-based messages, independently of the storage. This paper concentrates on two WFS mandatory operations: DescribeFeatureType and GetFeature. DescribeFeatureType returns an XML schema defining the feature type (i.e., the names and data types of the attributes that define the structure of the features in a given layer). GetFeature returns the instances of a certain feature type that match a geographical filter (e.g., return all roads within a geographic area). Both WMS and WFS have a GetCapabilities operation to return the service metadata (e.g., the layers list, supported response formats, supported versions, the URLs to invoke the other operations, etc.).

Internet Map Servers are server-side applications that implement and expose WMS and WFS services, among others.

B. ESB Mediation Patterns

Within an ESB-based platform, services and applications communicate by sending messages through the ESB. Messages are processed by mediations flows which can apply to them different mediation operations (e.g., routing). In this way, the ESB can ensure that applications and

services connect successfully [10][11]. Although mediations are not formally restricted in what they can do, there are a set of basic patterns, known as mediation patterns, that are seen repeatedly and have been documented [4][10][12][13].

Transformation patterns deal with the runtime transformation of messages. Routing patterns dynamically determines the message path according to different factors. For example, the Content-Based Routing (CBR) determines the message path based on its content. The Splitter Pattern breaks out a message into a series of individual messages. The Aggregator Pattern receives multiple messages and when a given set of messages is complete, a single message is returned consolidating their content. Finally, the Cache Pattern returns messages which were previously stored and returned as a response for the given request [13][14].

Figure 1 presents a summary of these mediation patterns and a graphical representation for them, introduced in [5].

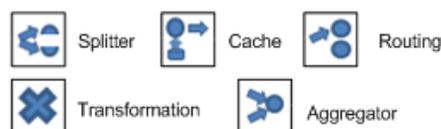


Figure 1. Summary of Mediation Patterns.

III. SOLUTION APPROACH

The proposed solution consists of an ESB-based Enterprise Integration Platform (EEIP) which provides reusable geo-oriented integration mechanisms to deal with common challenges of integrating OWS with enterprise applications. Figure 2 presents the general architecture of the platform and the external systems with which it interacts. The Basic and Complex Geo-oriented Integration Mechanisms are proposed and specified in this paper, while the underlying mediation mechanisms are usually available in ESB solutions. The platform acts as a broker between clients and servers (either OWS servers or enterprise systems). Clients send their requests to the ESB and the ESB routes them to real servers, applying some mediation flows, in a transparent fashion. These flows return value-added responses that could not be obtained by directly consulting the servers.

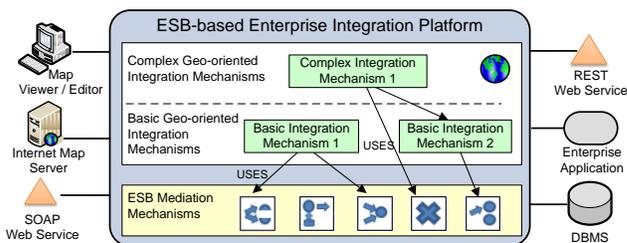


Figure 2. ESB-based Enterprise Integration Platform for Geospatial Web Services.

The ESB Mediation Mechanisms correspond to the mediation patterns described in section II.B (i.e., routing, splitter, etc.). These mechanisms are usually included in ESB products and provide reusable solutions to deal with general integration and communication requirements.

The Basic Geo-oriented Integration Mechanisms are higher-level mechanisms, built on top of the previous ones, which provide either geo-oriented utilities or reusable solutions to integrate OWS with enterprise systems.

The Complex Geo-oriented Integration Mechanisms are mediation mechanisms, built on top of the two previous ones, which provide higher level reusable solutions to integrate OWS with enterprise applications.

Additionally, the platform interacts with various external systems including IMSs, enterprise applications, SOAP / REST WS and map viewers / editors, among others. These systems provide information or consume services exposed in the platform. For instance, the platform can consume a SOAP WS to obtain business data from an enterprise system or a map viewer can consume a WMS interface exposed as a service in the platform.

IV. GEO-ORIENTED INTEGRATION MECHANISMS

This section presents and describes two complex geo-oriented integration mechanisms: WMS Enricher and SOAP-WMS Wrapper. Each mechanism is first described through a general description which includes: a motivation (i.e., why the mechanism is needed), a set of constraints (i.e., restrictions to be considered for the solution), a high level solution and an application example.

An in-detail solution is also provided for each mechanism. This in-detail solution is specified through mediation flows built on top of the ESB mediation mechanisms and some basic geo-oriented integration mechanisms, which are also described.

Finally, a set of variants (i.e., modifications or improvements) is analyzed for each mechanism.

A. WMS Enricher

The WMS Enricher is a complex geo-oriented integration mechanism which addresses the issues of integrating heterogeneous systems in GIS-based applications. To this end, it includes mediation flows to complement WMS responses with business data which are absent in the IMS.

1) General Description

TABLE I. presents the general description for the WMS Enricher mechanism.

2) In-detail Solution

Figure 3 presents a high level view of the WMS Enricher internals, which leverage a set of basic reusable geo-oriented mechanisms.

Instead of directly interacting with an IMS, a WMS client sends WMS requests to the EEIP. These requests are processed differently according to the operation involved (GetCapabilities, GetMap, GetFeatureInfo).

When the EEIP receives a WMS request, it creates an ESB message containing the request.

TABLE I. WMS ENRICHER GENERAL DESCRIPTION

WMS Enricher	
Motivation	An organisation has an IMS that is accessed by external WMS Clients. This organization wishes to complement its GI (accessible through the IMS) with related business data located in an enterprise system (accessible through a WS).
Constraints	Since the WMS Clients are external and the organization cannot modify them, the mechanism implementation must be transparent to those clients.
Solution	A mediation flow which enriches WMS responses is implemented within the EEIP. This flow is responsible for handling WMS requests, splitting them into business-dependant requests, forwarding these requests to the appropriate destination (IMS or enterprise system), aggregate the business-dependant responses and return the enriched WMS response to the client.
Application Example	The Ministry of Tourism has an IMS which publishes a Points of Interest (POI) layer that it wants to complement with the number of visitors in the last month of each POI. This information is provided by an independent company through a SOAP WS.

This message is first processed by the GeoEntryPoint mechanism (1), which obtains the invoked operation from the WMS request, set a message property with this operation and returns the modified message.

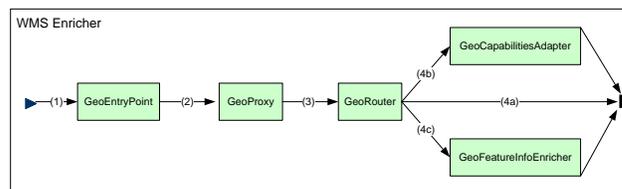


Figure 3. Aggregating data from WMS and other external systems.

Then, the message is processed by the GeoProxy mechanism (2), which works as a gateway to the backing IMS. It modifies the received message replacing the request with the response returned by the IMS.

Next, the message is processed by the GeoRouter mechanism (3). This mechanism routes the message according to the name of the invoked operation, which was previously stored in a message property.

If the operation is GetMap, the GeoRouter return the response to the invoking client (4a). If the operation is GetCapabilities, the GeoRouter routes the message to GeoCapabilitiesAdapter (4b). This mechanism is needed because the original GetCapabilities response contains the URLs to invoke the other operations (GetMap and GetFeatureInfo), and as a result, these have to be replaced so that clients can invoke them through the EEIP in the subsequent requests. If the operation is GetFeatureInfo, the response is enriched using other sources, for example, SOAP and REST WS. To this end, the GeoRouter routes the request to the GeoFeatureInfoEnricher mechanism (4c), which receives an ESB message containing a WMS response and, more precisely, the IDs of the features involved. The GeoFeatureInfoEnricher has the purpose of invoking the other external sources, using those IDs, to obtain additional

information of the features. It has to aggregate the responses obtained from these sources with the original IMS response.

In the following paragraphs, the basic geo-oriented integration mechanisms that have been mentioned are described in the depth.

GeoEntryPoint is a basic geo-oriented integration mechanism which receives ESB messages containing an IMS request, obtains data from this request and stores them in message properties for later processing. In order to use this mechanism, it is necessary to specify what information has to be obtained from the requests (e.g., invoked operation).

GeoProxy is a basic geo-oriented integration mechanism which has the purpose of invoking an OWS operation on an IMS. It receives a message containing a request and returns a modified message replacing the request with the response obtained from the IMS. In summary, this mechanism acts as an HTTP gateway. In order to use this mechanism, the URL of the IMS has to be specified.

GeoRouter is a basic geo-oriented integration mechanism which uses the ESB Routing, more precisely, the CBR. It routes messages to the appropriate destination according to message properties. To completely specify the behavior of this mechanism, a set of (property, value, destination) triples has to be specified.

GeoCapabilitiesAdapter is a basic geo-oriented integration mechanism which uses the ESB Transformation in order to replace all the original URLs published in the GetCapabilities response with the equivalent URLs that handle the requests through the EEIP. For instance, Figure 4 shows a fragment of a GetCapabilities response, in which the OnlineResource element indicates the URL where the GetFeatureInfo operation must be invoked using the HTTP Get method. Since this URL points to the IMS, GeoCapabilitiesAdapter must replace it with the corresponding URL in the EEIP (see Figure 5). To completely specify the behavior of this mechanism the required parameter is the list of pairs (original URL, adapted URL).

```
<GetFeatureInfo>
  <Format>text/plain</Format>
  <Format>application/vnd.ogc.gml</Format>
  <Format>text/html</Format>
  <DCPType>
    <HTTP>
      <Get>
        <OnlineResource xlink:type="simple"
          xlink:href="http://imshost:8080/ims/ows?SERVICE=WMS&" />
      </Get>
    </HTTP>
  </DCPType>
</GetFeatureInfo>
```

Figure 4. GetCapabilities response fragment showing the original URL of the GetFeatureInfo operation.

```
<GetFeatureInfo>
  <Format>text/plain</Format>
  <Format>application/vnd.ogc.gml</Format>
  <Format>text/html</Format>
  <DCPType>
    <HTTP>
      <Get>
        <OnlineResource xlink:type="simple"
          xlink:href="http://eeiphost:8080/eeip/ows?SERVICE=WMS&" />
      </Get>
    </HTTP>
  </DCPType>
</GetFeatureInfo>
```

Figure 5. GetCapabilities response fragment showing the adapted URL of the GetFeatureInfo operation.

GeoFeatureInfoEnricher is a basic geo-oriented integration mechanism which uses the ESB Splitter, Aggregator and Transformation in order to obtain business data from enterprise applications and consolidate the responses into a unique GetFeatureInfo response. Figure 6 shows the mediations flow that occurs inside this mechanism. As in [5], YAWL [15] is used to uniformly represent this mediation flow among ESB products.

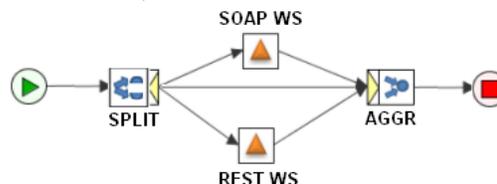


Figure 6. GeoFeatureEnricher Mechanism as a YAWL Net.

In the first place, a Splitter receives a message containing the IMS response and sends the requests to the external sources (e.g., a SOAP WS). To accomplish this task, some parameters have to be specified for each external source: its type (e.g., SOAP WS, REST WS, EJB, etc.), its address (e.g., an URL), the operation to invoke and the attribute name where the feature ID has to be set to invoke the operation. The Splitter has the knowledge to send the required information for each external source type. Also, the WS SOAP WS and REST WS mechanisms know how to build requests for each type of WS, respectively.

When all the responses are received, an Aggregator is in charge of consolidating them into a single GetFeatureInfo response. To accomplish this, some parameters have to be specified for each layer: the layer name, a set of triples of the form (attribute, source, locator) where *attribute* is the name of an attribute to be added to the response, *source* is an external source from where the attribute value is obtained, and *locator* is source-type dependant way to get the value from the external source response (i.e., an XPath or XQuery expression in the case of a SOAP response, etc.).

3) Variants

Following the same approach that has been applied for the WMS Enricher, a possible variant for this mechanism is to derivate a similar solution to enrich a non-transactional WFS, i.e., the WFS Enricher. In this scenario, a WFS client interacts with the EEIP issuing GetCapabilities, DescribeFeatureType and GetFeature requests. In the case of the WFS GetCapabilities operation, the same adaptation that is performed in the WMS Enricher applies without modification. In the case of DescribeFeatureType, the XML schema that is returned has to be augmented with the attributes that are not part of the original feature type. The GeoFeatureTypeAdapter mechanism is defined to accomplish this task. In the case of GetFeature, a similar mechanism to the GeoFeatureInfoEnricher is defined, the GeoFeatureEnricher, which queries the external applications to retrieve the additional data for each feature in the

GetFeature response. Figure 7 presents a high level view of this variant.

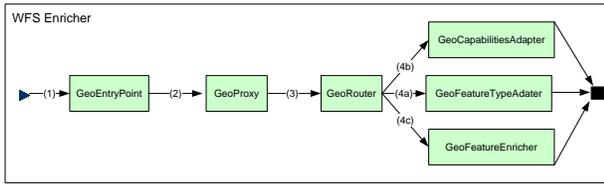


Figure 7. Aggregating data from a basic WFS and other external systems.

Given that this kind of integration solution has a clear impact on performance, another variant for the WMS and WFS Enricher mechanisms is to leverage previous processing to return a response. A possible strategy to do that is using previously returned information, through a cache mechanism [16]. In this case, the GeoEntryPoint could query the Cache before sending a request to the GeoProxy. If the response is found in the cache, it is returned to the client instead of following the usual flow.

B. SOAP-WMS Wrapper

The SOAP-WMS Wrapper is a complex geo-oriented integration mechanism which serves the purpose of publishing WMS services using W3C Web Services Standards (SOAP and WSDL). In this way, they can be integrated into general-purpose business-to-business (B2B) processes that rely on those standards and leverage an extensive stack of related specifications to address advanced concerns such as security, reliability, discoverability, orchestration, etc.(see discussion on [17]).

1) General Description

TABLE II. presents the general description for the SOAP-WMS Wrapper mechanism.

TABLE II. SOAP-WMS WRAPPER GENERAL DESCRIPTION

SOAP-WMS Wrapper	
Motivation	An organisation wishes to incorporate a IMS into its existing Web Services infrastructure, so as to make GI accessible to its existing business processes and applications.
Constraints	Since the organisation’s processes and applications are already implemented using SOAP WS and related WS-* specifications to meet some advanced constraints (e.g., WS-Security, WS-BPEL, etc.), it is not possible or desirable to consume WMS services using plain HTTP messages interchange.
Solution	A mediation flow which wraps WMS using SOAP WS is implemented within the EEIP. This flow is responsible for handling SOAP-WMS requests, translating those requests into standard WMS requests, forwarding these requests to the IMS, translating the WMS responses into SOAP-WMS and returning them to the client.
Application Example	The Ministry of Energy and Mineral Resources has implemented a BPMS, based on SOAP WS, to manage its business processes, which include awarding grants for mine prospectation, exploration and exploitation. In a first effort to incorporate GI into the system, an activity has been defined in an early stage of the process in which an analyst can visualize the

SOAP-WMS Wrapper	
	mine area affected by the grant request and the protected areas layer (where mining is prohibited) on the same map.

2) In-detail Solution

The SOAP version of WMS is published in the EEIP following the guidelines from [18]. Figure 8 shows a high level view of the SOAP-WMS Wrapper.

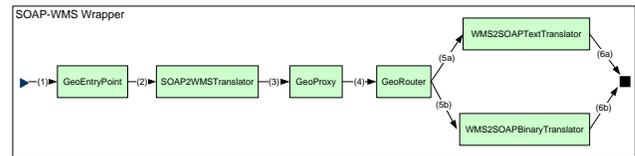


Figure 8. Wrapping WMS into SOAP WS.

Once a SOAP request arrives at the EEIP an ESB message is created containing this SOAP request. The message is first processed by the GeoEntryPoint mechanism (1) which obtains, from the request, the operation that is being invoked and stores its name in a message property. The modified message is then processed by the SOAP2WMSTranslator (2) which has to decode the SOAP request, create a standard WMS request, either using the KVP or XML encoding, and replace, in the message, the SOAP request with a standard WMS request. The output message is then processed by the GeoProxy (3) which invokes the operation on the IMS, receives the result and replaces the request with the response. The ESB message is then processed by the GeoRouter (4) according to the operation that was invoked. If the operation returns a text response (i.e., GetCapabilities and GetFeatureInfo), it is sent to the WMS2SOAPTTextTranslator (5a), if it returns a binary response (i.e., GetMap) it is sent to the WMS2SOAPBinaryTranslator (5b). These two mechanisms work in the same way, by decoding the response and creating a SOAP message to return to the client (6a and 6b).

SOAP2WMSTranslator is a basic geo-oriented integration mechanism which uses the ESB Transformation in order to convert a SOAP message into a WMS request. This mechanism can be configured to either generate a KVP o XML encoded requests, to be used by the HTTP Get or HTTP Post methods respectively [19]. The Transformation mechanism is dependent on the ESB platform of choice, but typically includes XML-based XSLT transformations or object marshalling/unmarshalling, the latter being programming language bound. To completely specify the behavior of this mechanism the required parameters are the type of encoding (KVP, XML) and the URLs of the IMS that handle each type of request.

WMS2SOAPTTextTranslator is a basic geo-oriented integration mechanism which uses the ESB Transformation in order to convert a WMS response into a SOAP message. Only text-format responses are processed by this mechanism, i.e., the responses of the GetCapabilities and GetFeatureInfo operations.

WMS2SOAPBinaryTranslator is a basic geo-oriented integration mechanism which uses the ESB Transformation in order to convert a WMS response into a SOAP message. Only binary-format responses are processed by this mechanism, i.e., the responses of the GetMap operation. This mechanism uses the Message Transmission Optimization Mechanism (MTOM) to attach the binary data to the SOAP message.

3) Variants

An analogue mechanism to the SOAP-WMS Wrapper, could make possible to expose other OWS's as SOAP WS. In the case of WFS, its inclusion in a Service Oriented Architecture (SOA) using the EEIP could go even further than just visualizing GI, as is the case with WMS, since WFS offers a complete interface to perform advanced queries and transactions on GI.

V. IMPLEMENTATION DETAILS

In order to show the feasibility of the proposed approach, some prototypes have been developed based on the JBoss ESB product, using GeoServer as the IMS. These prototypes have also allowed identifying and analyzing key implementation aspects.

In particular, the WMS Enricher mechanism was developed by leveraging various built-in features of JBoss ESB, like its CBR and aggregator features [20]. The variants of the WMS Enricher mechanism (WFS Enricher and Cache) were also implemented [21] leveraging these features. However, some extra work was required to implement a cache mechanism, given that JBoss ESB does not natively provide this feature.

Finally, the WMS SOAP Wrapper was also successfully implemented with JBoss ESB [22] leveraging its native features, like a SOAP Processor to perform the marshalling and unmarshalling.

VI. CONCLUSION AND FUTURE WORK

This paper addressed the issues of integrating GIS with enterprise systems to build large-scale Information Systems that use GI in a broader business context. More concretely, it presents an ESB-based Integration Platform which provides generic Geo-oriented Integration Mechanisms to facilitate the integration of GIS, and more specifically OWS, with traditional enterprise counterparts. Those mechanisms are described and specified in detail through mediation flows, which extend mediation patterns commonly supported in ESB products (e.g., Splitter and Aggregator).

Also, the development of various prototypes has shown the feasibility of the proposed approach and mechanisms.

The main contributions of this paper consists of the specification and implementation of a reference platform and concrete mechanisms, based on a general purpose ESB, that address key aspects of integrating heterogeneous systems in a GIS environment. The specification and implementation of this kind of platform has not been tackled by previous works to the best of authors' knowledge.

As part of an ongoing project, this work aims to be a step forward in developing a comprehensive platform which

facilitates organizations the task of integrating GIS with their traditional enterprise systems.

Future work consists in improvements to these mechanisms, the design and development of new ones (e.g., a SOAP-WFS Wrapper, a Transactional WFS Enricher, etc.), and the exploration of approaches to carry out the integration with an ESB-based e-Government platform.

REFERENCES

- [1] D. J. Maguire, "An overview and definition of GIS" in D. J. Maguire, M. F. Goodchild, & D. W. Rhind, Eds., *Geographical Information Systems: Principles and Applications* (Vol. 1), Harlow, UK: Longman Group, pp. 9-20, 1991
- [2] M. S. Witkowski, P. M. Rich, G. N. Keating, and L. A. N. Laboratory, "A prototype for enterprise GIS", Los Alamos National Laboratory, 2004, electronic resource.
- [3] B. Rienzi, R. Sosa, P. Foti, and L. González, "Benefits and challenges of using geographic information systems to enhance social security services," presented at the 6th International Policy and Research Conference on Social Security, Luxembourg, 2010.
- [4] D. Chappell, "Enterprise Service Bus: Theory in Practice", O'Reilly Media, 2004.
- [5] L. González and R. Ruggia, "Addressing QoS issues in service based systems through an adaptive ESB infrastructure", *Proceedings of the 6th Workshop on Middleware for Service Oriented Computing - MW4SOC '11*, pp. 1-7, Lisbon, Portugal, 2011.
- [6] A. Whiteside, Ed., "OpenGIS Web Services Architecture Description", Open GIS Consortium Inc., 2005.
- [7] J. de la Beaujardiere, Ed., "OpenGIS Web Map Server Implementation Specification", Open GIS Consortium Inc., 2006.
- [8] C. Kottman, C. Reed, Eds., "The OpenGIS Abstract Specification. Topic 5: Features", Open GIS Consortium Inc., 2009.
- [9] P. A. Vretanos, Ed., "OpenGIS Web Feature Service 2.0 Interface Standard", Open GIS Consortium Inc., 2010.
- [10] M. T. Schmidt, B. Hutchison, P. Lambros, R. Phippen, "The enterprise service bus: making service-oriented architecture real", *IBM Syst. J.* 44, 781-797, 2005.
- [11] M. Papazoglou, "Web Services: Principles and Technology", Prentice Hall, 2007.
- [12] H. Wylie, P. Lambros, "Enterprise Connectivity Patterns: Implementing integration solutions with IBM's Enterprise Service Bus products", <http://www.ibm.com/developerworks/library/ws-enterpriseconnectivitypatterns/index.html>.
- [13] G. Hohpe and B. Woolf, "Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions", Addison-Wesley Professional, 2003.
- [14] R. Yan Fang, F. Ru, T. Zhong, L. Eoin, S. Harini, T. Banks and L. He, "Cache mediation pattern specification: an overview", <http://www.ibm.com/developerworks/webservices/library/ws-soa-cached/>.
- [15] A.H.M. ter Hofstede, W.M.P. van der Aalst, M. Adams and N. Russell: *Modern Business Process Automation: YAWL and its Support Environment*. Springer, 2009.
- [16] L. González and R. Ruggia, "Towards dynamic adaptation within an ESB-based service infrastructure layer", *Proceedings of the 3rd International Workshop on Monitoring, Adaptation and Beyond*, pp. 40-47. ACM, New York, NY, USA, 2010.

- [17] J. T. Sample, K. Shaw, S. Tu, and M. Abdelguerfi, Eds., "Geospatial Web Services: Bridging the Gap Between OGC and Web Services," in *Geospatial Services and Applications for the Internet*, Springer US, 2008, pp. 72–92.
- [18] J. Sonnet and C. Savage, "OWS 1.2 SOAP Experiment Report", Open GIS Consortium Inc., 2003.
- [19] R. Gartmann and B. Schäffer, "OpenGIS Wrapping OGC HTTP-GET and -POST Services with SOAP - Discussion Paper", Open GIS Consortium Inc., 2008.
- [20] E. Rosa, J. Davison and G. Schnyder, "Integración de Datos Empresariales en Servicios Geográficos a través de un ESB", TSIG, 2010, unpublished,
<http://www.fing.edu.uy/inco/cursos/tsi/TSIG/papers/wms-enr.pdf> [accessed: 2012-12-21]
- [21] E. Penna, V. Perez and M. Steffen, "WFS en ESB con Fuente Externa y Cache", 2011, unpublished,
<http://www.fing.edu.uy/inco/cursos/tsi/TSIG/papers/wfs-enr.pdf> [accessed: 2012-12-21]
- [22] P. García, J. Laborde and M. Rodríguez, "WMS SOAP con ESB", TSIG, 2011, unpublished,
<http://www.fing.edu.uy/inco/cursos/tsi/TSIG/papers/wms-soap.pdf> [accessed: 2012-12-21]