WS-GIS: Towards a SOA-Based SDI Federation

Fábio Luiz Leite Jr., Cláudio de Souza Baptista, Patrício de Alencar Silva, Elvis Rodrigues da Silva

Information Systems Laboratory, University of Campina Grande
Av. Aprigio Veloso, 882, 58109-970 Campina Grande – Pb, Brazil
{fabio, baptista, patricio, elvis}@dsc.ufcg.edu.br

Abstract. *The volume of spatial data available on the Web has been increasing rapidly. Currently, the Web has been used not only for document searching but also for the provision and use of services, known as Web services, which are published in a directory and may be automatically discovered by software agents. Particularly in the spatial domain, the possibility of accessing these large spatial datasets via Web services has motivated research into the new field of Spatial Data Infrastructure (SDI) implemented using service-oriented architecture. In this paper, we propose WS-GIS which is a SDI using Web services. WS-GIS enables spatial search on distributed catalogs that form a federation of spatial databases.*

1. Introduction

The amount of spatial digital information that has been generated in the last few years is increasing rapidly due to many reasons. Firstly, in the field of Earth Observation Systems (EOS), huge projects from space agencies such as NASA and CEO are gathering petabytes of information per year in satellite images which have been accumulated with current stored data. Secondly, map agencies are producing their maps electronically. Lastly, advances in mobile devices and wireless infrastructure have motivated the large use of spatial related data.

There are several issues to be addressed in order to make these large datasets easily accessed. Firstly, these data should be indexed and retrieved by spatial-aware search engines, as current search engines use only text-based retrieval, which is unsuitable for spatial data [Fu et al. 2005]. Secondly, data replication is an important issue and should be minimized, as dealing with spatial data involves high costs not only in terms of acquisition, extraction, transformation and loading but also storage and maintenance. Finally, the use of standards is a mandatory issue, as they enable both data and services to be easily discovered and to interoperate. Hence, the implementation of a spatial data infrastructure – SDI – at global, national, and local scale, using largely adopted standards based on service-oriented architecture is a very hot research topic [Davis and Alves 2005, Georgiadou 2006].

One of the first attempts of providing access to these spatial datasets was the design of clearinghouses. The main aim of a clearinghouse was to provide a centralized Web portal in which, by using metadata annotation on spatial data distributed over the Web, users may search, view and transfer such data [Crompvoets et al. 2004]. Data providers should then subscribe into the clearinghouse in order to make their spatial data retrievable.

More recently, the advent of service-oriented architecture has motivated the adoption of geoportals, which consist of Web portals based on services, data, search engines and applications. These geoportals provide access to data and metadata and links to the service providers. Thus, there is a shift from a data-centric SDI to a service-oriented one [Maguire and Longley 2005].

Nowadays, the Web is used not only for document searching but also for the provision and use of services, known as Web services [Alonso et al. 2003], which return dynamically changing data. Particularly in the spatial domain, examples of such services include but are not limited to basemaps, coordinate transformation, gazetteer, location-based systems, routing, overlay, buffer, and so on [Davis and Alves 2005].

In the Web services domain these standards involve a common protocol for exchanging messages; a repository which enables to catalog and to search for services; a description language for service annotation and a mechanism of service composition.

Several of such standards have been proposed. The SOAP (Simple Object Access Protocol) protocol specifies how Web services may encapsulate messages in XML documents. WSDL (Web Service Description Language) aims to describe service interfaces offered by Web services, and it may describe service operations, input and output parameters of each message, data type parameters and so on.

UDDI consists of a framework which enables the publication and discovery of services. Service providers use UDDI to announce their services; whereas clients may use it in order to find out a specific service, and how to interact with it. These standards are based on XML.

Currently there are several Web services published on the Internet, however, an important issue is how to find out precisely the service a user is requesting? Keyword-based search techniques implemented by search engines are not adequate for service discovery. Some reasons for this limitation include [Alonso et al. 2003]:

- UDDI registry contains little textual information about the underlying services. Usually there is just a short description of the service provider;
- Keyword-based search may retrieve many irrelevant services just because they contain the addressed keyword. Also, a relevant service may not be retrieved just because this service uses synonyms or related terms to the used keywords.
- It is very complex to describe a given service based only on keywords;
- WSDL and UDDI descriptions express almost none of the service semantics;

Nonetheless, when dealing with spatial data these directories should become spatial-aware. For example, a user might be interested in a map with rivers inside a particular region.

The lack of semantics in SOA has been addressed recently [Burstein et al. 2005, Sivashanmugam et al. 2004]. One possible solution is to enhance the catalogs with semantic through the use of ontologies, that represents a formal specification of a shared conceptualization [Gómez-Perez et al. 2004]. There are some proposals on the use of spatial domain ontologies to improve resource discovery on the Web [Fu et al. 2005].

In this paper we propose a service-oriented architecture for spatial data sharing. Hence, users may query a distributed catalog service, semantically enhanced with ontologies. The remainder of the paper is organized as follows. Section 2 presents our architecture. Section 3 addresses some implementation issues through a real example. Section 4 highlights related work. Finally, section 5 concludes the paper and discusses further work that should be undertaken.

2. The WS-GIS Architecture

We propose WS-GIS which is a SDI based on service-oriented architecture. WS-GIS enables spatial search on distributed catalogs that form a federation of spatial databases. The architecture (see figure 1) enables access to spatial Web services and it has the following features:

- Resource location capability in federated catalogs: by using ontologies it is possible to model catalog services, so that distributed searches, even spatial, on such catalogs may be executed. For instance, a fireman may pose the following query, when faced to a forest fire: "Which kind of vegetation and animal species might be affected by the fire spots identified in a particular satellite image?";
- Query caching: previous services chaining may be stored in the workflow for future reuse;
- Domain transparency: client should not care about where the resources come from.

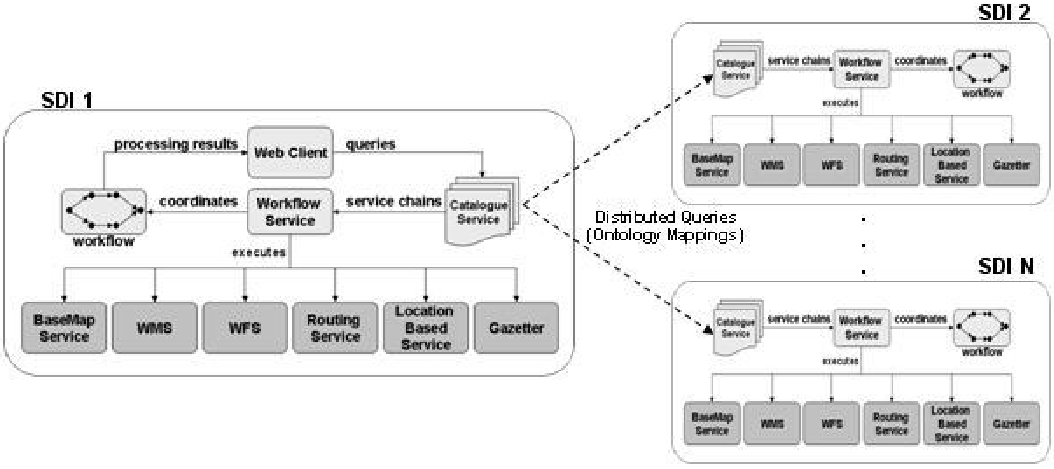

Figure 1. WS-GIS overall architecture

In figure 1, catalog service represents an OGC Catalog service which describes the underlying services using both services and domain ontologies. WFS and WMS represent the OGC Web Feature Service and Web Map Service, respectively [Open Geospatial Consortium 2006]. LBS is a Location-based service for context-aware applications [Almeida et al. 2006], the routing service is responsible for wayfinding and the Gazetteer service implements a gazetteer based on [Pazinatto et al. 2002]. Each SDI contains a set of spatial services provided by a given public or private organization. The SDIs cooperate through a catalog federation. Hence, it is possible to obtain resources provided by any SDI in the federation from any client. In the proposed architecture, the client agent might be able to query the client SDI catalog; which may distribute the query to other SDIs and it is responsible for composing the query results and sending it to the client. In the following each of these services are further detailed.

2.1. The Catalog Service

The catalog service is the core of our architecture, as it is responsible for SDIs integration at local, national or international scope. Each SDI maintains a spatial-aware catalog which may find resources locally. The catalog implements a communication protocol which propagates rewritten queries to other SDI catalogs.

The catalog communication protocol may choose one catalog service to cumulate query results, which contain references to the services, and send them to the workflow service. We use the following strategy to maintain the federation:

- when a registry enters into the federation, the administrator must insert into its registry an URL which already is part of the federation;
- after getting a registry address in the federation, the new registry sends a message to obtain the address table of the contacted site.

After that, all new addresses obtained by the new registry should be transmitted to the machines which are part of its address table. If there is no such new address, it will send an acknowledgment which confirms its membership in the federation.

In order to comply with OGC standards, the catalog service implements an OGC Catalog API which contains getCapabilities, getRecords, getRecordById, describeRecord, getDomain, presentResults and transaction methods.

In the following the registry and query modules are presented, followed by a discussion on how these catalogs semantically interoperate.

2.1.1. The Registry Module

The catalog maintains a metadata repository which describes the SDI underlying resources. These metadata are implemented using both service ontologies, which provide service semantics; and domain ontologies, which describe data semantics. After receiving a request, the catalog queries the metadata repository in order to find the services which fulfill the request. The catalog represents these ontologies using the Jena framework which stores them in a relational database system. The underlying data is stored in a spatial-aware database. This registry module uses the Jena framework to manipulate the ontologies.

2.1.2. The Query Module

The query module retrieves information based on query spatial features. Firstly, this module retrieves metadata that describe the available services and then, it performs spatial operations based on user requirements. Usually, the geographic information provided in the query is based on place names. Thus the catalog may search for a Gazetteer service which may return the geographical coordinates of a given place name.

The query module also makes a query broadcast to the other catalogs in the federation. Hence, the catalog service attempts to answer user query only regarding local catalog data. In the case of partial fulfillment of user requirements, the catalog propagates the query to other catalog instances. The query propagation algorithm follows:

```
Algorithm search (query, hops) {
  if (localOntology != query.ontology)
     query = mapping (localOntology, query);
     // semantic matching among the SDI catalogs
  if (query is fully satisfied)
     return resultSet; // as service references
  else {
     resultSet = partialResult();
     newQuery = mapping (globalOntology, (query – resultSet));
     if (hops threshold is not achieved) {
        resultSet = resultSet ∪ otherSDIs.search (newQuery, hops);
        // If a timeout is achieved throws an exception
        return resultSet;
     } else
        // throws an exception;
  }
}
```

2.1.3. Semantic integration among catalogs

Ontologies are being applied very successfully in supporting information and knowledge exchange. However, for many reasons, different people and organizations will tend to use different ontologies. Unfortunately, the semantic heterogeneity also represents a drawback when different network catalogs make use of different domain ontologies. Hence, the system should make ontology mappings between the SDI ontologies, trying to reconcile the concepts and properties related to them. It is important to mention that so far there is no provision for semantics in service and data discovery in OGC specifications.

Each SDI defines its own ontology that is used in its catalogs. It means that there is a common vocabulary in each SDI that represents the concepts of the domain ontology used to describe the resources. However, we make an assumption that there is a set of concepts which can be shared among the SDIs, through clustering semantic communities. Thus, communities are formed through semantic mapping between SDIs. Each community has its own domain ontology which can be shared, and the other communities can cooperate in this scenario, mapping their domain ontologies. This can contribute to simplify the semantic mappings related to heterogeneity among SDIs [Hameed et al. 2004].

Considering this scenario, the query module broadcasts the rewritten query in the global SDI ontology. Thus, the catalogs that receive the propagated query make the semantic mapping between the local ontology and the global one. To accomplish this task it is necessary to verify in which relationship level these mappings are possible. Once the referred catalogs belong to the same community, we can make the assumption that they share the same ontology, which is the global one. In the context of this work, we make an assumption that all the communities reconcile their ontologies. Through the mapping files originated from the ontology reconciliations, the catalogs share the concepts used by the SDIs and interprets the network propagated queries [Hameed et al. 2004].

2.2. The Workflow Service

The workflow service is also responsible for executing the selected service chaining. The catalog service returns a record to the user that contains all the possible workflows that are able to fulfill the submitted request. The next step is the selection of the desired service workflow by the user. Then, the catalog service activates the workflow service, providing the workflow selected by the client as parameter. The workflow service executes the selected workflow and accomplishes the monitoring phase that comprises basically aspects like checking the right functioning of the involved services, response time, service availability, unexpected behavior and error handling. Furthermore, for each service involved in the workflow, the service inputs and outputs are checked so that each output of a service is compatible with the input of the subsequent service in the workflow.

2.3. The LBS Web Service

Mobile users may register to receive alerts on subject of their interest while they are on the move. The LBS Web service is responsible for receiving and managing user context information. When a client is registered and provides profile, and appointments, it is able to receive context-aware information. The main available operations are:

- registerUser() which registers a user in a Web service, its profile and context information;
- updatePosition() which enables to update user geographic location in the system, and analyses the context searching for actions which are relevant to user context. Each time user location is updated the context is analyzed;
- registerAppointment() which registers a user appointment.

2.4. The Web Map Service and Web Feature Service

These services are based on the OpenGeospatial WMS and WFS. Nonetheless, we have implemented a Web service interface for such services. A request is done through several parameters in a URL. By implementing these services as Web services we not only make the requests easier, but also it is possible to publish these services in a directory services such as UDDI, in order to provide automatic service discovery and invocation.

Currently, we have implemented the following WMS operations: *getCapabilities*, *getMap* and *getFeatureInfo*. The *getCapabilities* method obtains service metadata which describes relevant information including maximum number of layers which may exist in a map, available layers and styles, coordinate reference systems (CRS), bounding box and scale. The *getMap* method returns a map according to the parameters received. These parameters include Layers, which contains the list of layers in a map; CRS, which contains the coordinate reference system; BBOX, which expresses the bounding box of interest; and Format, which contains the map output format. The getFeatureInfo method enables to retrieve more information about a chosen feature, for instance, a River name, length, quality of water, etc.

The Web Feature Service specification enables the user to query and update geospatial data in an interoperable way. This service contains the following operations: *getCapabilities*, *describeFeatureType* and *getFeature*. The *getCapabilities* method is responsible for describing available features, and the operations supported by each feature. The *describeFeatureType* method obtains the structure of a given feature type, which is described in an XMLSchema. Finally, the *getFeature* method enables users to specify which feature properties will be queried, as for example the name of a given feature.

2.5. The Routing Service

This service is responsible for providing on demand routes among two or more places. The client sends a request to this service (using the getRoute() method) providing two pairs of latitude and longitude coordinates. The client may also inform some preferences to be taken into account by the service such as road and traffic conditions.

The routing Web service stores data about routes and their intersections as a graph. The path between each two pairs is pre-computed and stored in the database, using the total materialization strategy. The storage cost using this approach is high, but as disk prices are decreasing, the gains obtained in processing power are worthy. We use a graph hierarchy to minimize the storage costs. The graph is partitioned into non-interleaving sub-graphs and there are nodes which take part in more than one sub-graph, they are called border nodes and they take part in the super-graph.

Paths are computed in each fragment and in the super-graph. We used the classical Dijkstra best path algorithm to pre-calculate all costs [Dijkstra 1959].

2.6. The Gazetteer Service

Gazetteer is used to help users to find bounding box coordinates from place names or feature types, and vice-versa. The simplest way of implementing a gazetteer is by having pairs of place names and spatial footprints. In this case, the gazetteer does not consider the feature type of each element. The feature types are either implicit or explicit. In the former case, it is assumed that the user is familiar with the place name type, for example, England is a country. In the latter, the feature type comes together with the place name, as, for example, North Sea.

A reasonable core gazetteer should include feature type together with place name and footprint. This enables determining, for example, whether bank is a building or a land formation along the edge of a river [Pazinatto et al. 2002].

We have implemented a Gazetteer Web service which contains place names, spatial footprints, temporal interval, feature types, and lineage. The first two attributes are mandatory, whilst the latter three are optional. Place name contains the name of the place being identified; spatial footprint contains a bounding box represented by two pairs of latitude/longitude points; time is an interval that contains two timestamps: the beginTime and endTime of that place name; feature type contains a feature that characterizes the place name; and lineage contains the source that provided that information.

This service is accessed via searching one of the three attributes: place name, space and feature type. For example, a user may be interested in where a determined place is, what exists in a determined spatial footprint, or where a particular feature type exists on the Earth's surface.

2.5 Implementation issues

The Web services were implemented using the Java J2EE and JAX-RPC. Each service was developed in three steps: interface definition, which contains the methods which may be called by a client; interface implementation, which contains the service logic; and the service deployment in an application server. We have used the Java Sun Application Server. After the second step is done, the WSDL file is generated automatically.

All the implemented data services (WMS, WFS) and the processing services (LBS, routing services) use PostgreSQL with Postgis DBMS. The catalog service also uses PostgreSQL DBMS to materialize the ontologies and the Jena framework as inference engine to verify the ontology consistence.

3. An Example Scenario for the WS-GIS

In the Brazilian State of Paraíba there are many state departments and agencies which deal with data on natural resources and infrastructure. They manipulate information about water resources, weather, roads, etc. Unfortunately the underlying information systems do not interoperate and there is too much replication. Obviously this replication introduces inconsistency and results in a waste of space on disk, especially when dealing with spatial data such as satellite images and vector maps which require huge space on disks. Let us suppose that there exist three departments, each one with its SDI:

- the Water Management Agency (AESA), which deals with rivers, dams, lakes, wells, and so on;
- the Environmental Management Agency (SUDEMA), which deals with quality of water (rivers, dams, lakes, wells and so on); and
- the infrastructure agency (DER) which deals with road maintenance.

In these SDIs there is data replication: bodies of water are replicated in SUDEMA.

Suppose that a user would like to visualize a SVG (a W3C Specification based on XML to display graphics on a browser) map which contains the regions of Paraíba, the rivers (and its water quality) in Sertão region, and roads of the entire State, even whether these data layers are distributed in many agencies. Obviously, transparency on distribution is a mandatory issue [Garcia-Molina et al. 2001]. Moreover, let us suppose that the AESA SDI contains a Gazetteer service and a MapConvertion service which receives a GML document and returns it in SVG.

The user interaction follows. Firstly, the client executes, for example in the AESA SDI, the getRecords method, with OGC Filter parameters, in the catalog service requesting the desired information (e.g. a map with regions of Paraiba, rivers and information on water quality in Sertão region and roads). In order to pose the query the user uses concepts defined in the ontology which pertains to the local SDI. After receiving the query, the AESA catalog service proceeds as follows, according to the algorithm presented in section 2:

(a) it queries the Gazetteer service of the local SDI, in order to obtain the State region geometries to query the service registries using the intersection with rivers and roads;

(b) it verifies in the local SDI if it contains services with the requested data. In order to do that, the catalog runs a spatial query searching for the rivers in the Sertão region, water quality data and roads;

(c) as the AESA SDI does not have the whole requested data set, the query is propagated to other SDI catalogs. However, the local catalog rewrites the original query in order to remove from it the data which is already found in its database. Then, the catalog rewrites the query searching for roads and quality of water data in the geographic area of interest and propagates this query to other catalogs.

```
<ogc:Filter>
  <ogc:Or>
    <ogc:And>
     <PropertyIsEqualTo>
      <PropertyName>FeatureName</PropertyName>
      <Literal>River</Literal>
     </PropertyIsEqualTo>
     <PropertyIsEqualTo>
      <PropertyName>hasQuality</PropertyName>
      <Literal>yes</Literal>
     </PropertyIsEqualTo>
     <Overlaps>
       <PropertyName>Geometry</PropertyName>
       <gml:Polygon srsName="urn:ogc:def:crs:EPSG:6.6:4618">
        <gml:exterior>
         <gml:LinearRing>
          <gml:posList>
      -37.001821,-7.175744     -36.992499,-7.180127     -36.976542,-
7.160474 ... -37.019035,-7.190465 -37.001821,-7.175744
           </gml:posList>
          </gml:LinearRing>
        </gml:exterior>
       </gml:Polygon>
     </Overlaps> </ogc:And>
    <ogc:And>
     <PropertyIsEqualTo>
      <PropertyName>FeatureName</PropertyName>
      <Literal>Road</Literal>
     <PropertyIsEqualTo>
      <gml:posList>
       -38.76516,-6.91093 -38.75917,-6.9155 -38.75912,-6.91657 ...
-38.75622,-6.9036 -38.76516,-6.91093
      </gml:posList>
    </ogc:And> </ogc:Or>
</ogc:Filter>
```

It is important to notice that, when the SUDEMA catalog receives the query, it needs to perform the ontology matching of the concepts which come from AESA with its own ontology concepts, as they probably use different ontologies. Hence, there is a State ontology which has mappings for each agency ontology. Figure 2a presents part of AESA ontology which is the same of the State one. Figure 2b presents the SUDEMA ontology. It is important to notice that in the AESA ontology, and then in the State one, the body of water class is superclass of dam, river and well classes. Nonetheless, in the SUDEMA ontology there is no river, dam and well concepts, all of them are treated in a unique class called body of water.

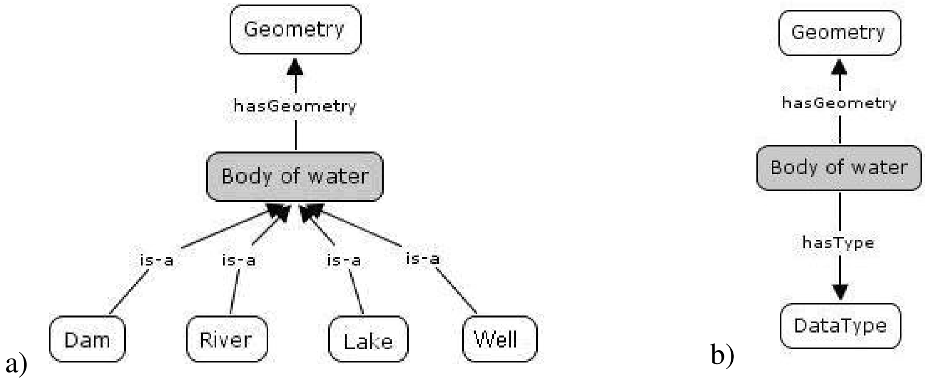

Figure 2. a) AESA and State ontology; b) SUDEMA ontology

In the following there is an example of ontology mapping between SUDEMA and State using OWL. Thus, the concept body of water in SUDEMA is mapped directly to the one in the state ontology. Hence, the "river" concept in AESA ontology (and in the state) is mapped to the "body of water" concept in the SUDEMA ontology through "hasType" property, defined as "river" DataType.

```
<owl:Class rdf:ID="BodyOfWater1SUDEMA">
   <owl:equivalentClass
rdf:resource="BodyOfWaterAESA"/>
</owl:Class>
```

The following OWL code indicates that the SUDEMA BodyOfWater class is the union of lake, river, well and dam in the State ontology, and consequently in the AESA one.

```
<rdf:RDF xmlns="http://localhost/ontologias/sudema.owl#"
     xml:base="http://localhost/ontologias/sudema.owl"
     xmlns:globalSDI="http://localhost/ontologias/globalSDI.owl#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:dc="http://purl.org/dc/elements/1.1/"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:owl="http://www.w3.org/2002/07/owl#">
```

```
    <owl:Ontology rdf:about="">
        <owl:imports
rdf:resource="http://localhost/ontologias/aesa.owl"/>
    </owl:Ontology>
    <owl:Class rdf:ID="Body_of_Water">
        <rdfs:subClassOf rdf:resource="&owl;Thing"/>
        <rdfs:subClassOf>
            <owl:Class>
                <owl:unionOf rdf:parseType="Collection">
                    <owl:Class rdf:about="&globalSDI;Dam"/>
                    <owl:Class rdf:about="&globalSDI;Lake"/>
                    <owl:Class rdf:about="&globalSDI;Well"/>
                    <owl:Class rdf:about="&globalSDI;River"/>
                </owl:unionOf>
            </owl:Class>
        </rdfs:subClassOf>
    </owl:Class>
    <owl:DatatypeProperty rdf:ID="hasType"/>
    <owl:Class rdf:ID="Geometry"/>
</rdf:RDF>
```

In parallel, the DER catalog also receives the request from AESA, and it solves the semantic differences between concepts and verifies in its metadata repository that it contains the Paraíba roads. Thus, it rewrites the query using the State ontology and broadcasts it to other SDI (in this case, the query concerns only quality of water). Lastly, it returns the references of the data services to the AESA catalog.

After receiving all services references, the query module mounts a service chaining and sends it to the client. This workflow is presented in figure 3.

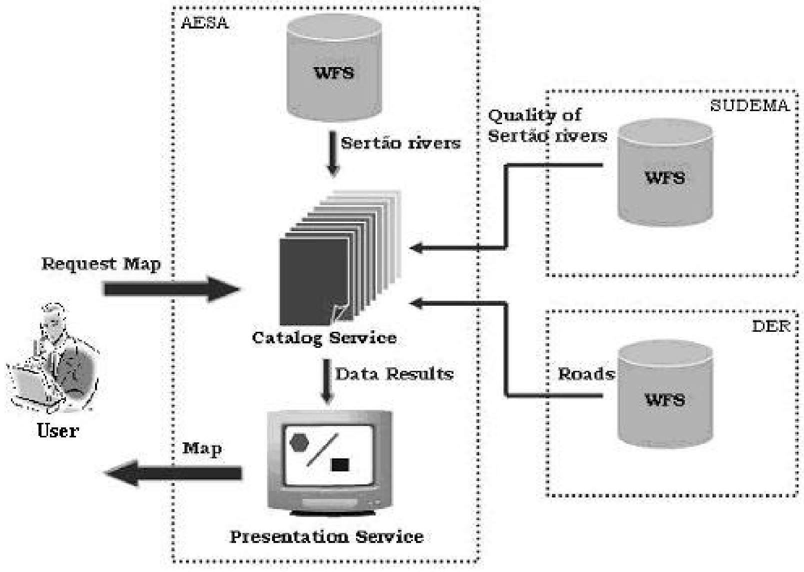

Figure 3. Service chaining of the query

The workflow service executes the service chaining and monitors the behavior of the involved services. Also, it sends the inputs and outputs necessary to each service. The final result is returned to the client which may render the map as it is presented in figure 4, using the framework iGIS [Baptista et al. 2004].

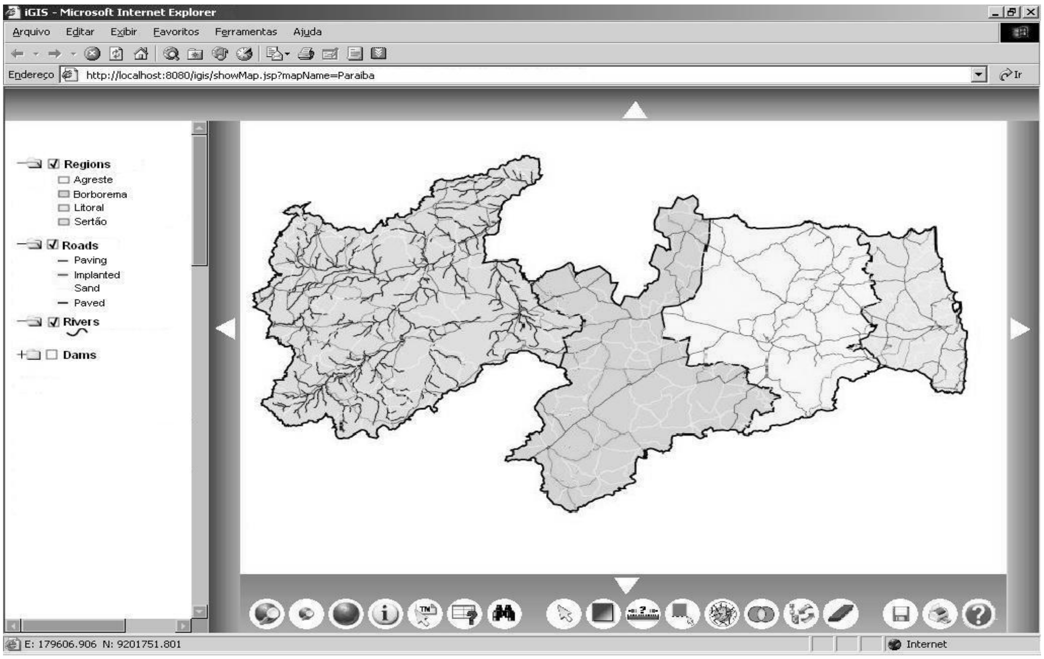
iGIS - Microsoft Internet Explorer
Arquivo Editar Exibir Favoritos Ferramentas Ajuda
Endereço http://localhost:8080/igis/showMap.jsp?mapName=Paraiba
Ir
☑ Regions
Agreste
Borborema
Litoral
Sertão
☑ Roads
Paving
Implanted Sand
Paved
☑ Rivers
□ Dams
E: 179606.906 N: 9201751.801
Internet

Figure 4. Query result on a map tool

4. Related Work

Bernard and Craglia [Bernard and Craglia 2005] present a survey on SDI and discuss its evolution from clearinghouses to geoportals. They emphasize the importance of migrating to a service-oriented infrastructure, in which Web services play a central role. Moreover, efforts on standards for these services are highlighted. These efforts result in a (GI)-Service Driven Infrastructure which characterizes the next generation of SDI which is focused on distributed and shared functionalities. Also, new functionalities can be achieved by service chaining.

The provision of service chaining enables scalability without compromising performance. There are several issues which should have been taken into account such as the level of transparency of the chaining to the client and the effort required in the client to implement service coordination. According to Alameh [Alameh 2003], there are three types of service chaining:

- Client-Coordinated service chaining: the client defines and controls the service execution order in the chaining;
- Static Chaining using aggregate services: the service chaining is hidden from the client, so the latter has only a link to the chaining. However the chaining is not dynamic;

- Workflow-managed service chaining with mediating services: combines the simplicity of static chaining using aggregate services with the flexibility and control provided by the client-coordinated service chaining. The workflow service is responsible for error manipulation and exceptions, and for mounting dynamically the service chaining to be used by clients.

Although, Alameh's proposal deals with service chaining, there is no concern on locating spatial resources in distributed catalogs.

Zhao et al. [Zhao et al. 2004] propose an integration model which uses a Metadata CatalogService (MCS) for storing metadata and retrieving resources in grids; and the model OGC ebRIM, which enables to aggregate semantics to the spatial queries in distributed architecture based on grids. Their idea is to implement semantic mapping between these two models by using ontologies in OWL. Therefore, user poses queries at the semantic level and those queries are mapped into the grid through MCS. Nonetheless, this work does not address the mapping between OGC Catalog Services – ebRIM and the UDDI registries, which underpin the service-oriented SDIs.

Sivashanmugam et al. [Sivashanmugam et al. 2004] propose a peer-to-peer network of public and semi-private UDDI registries, which enables transparent access to any registry in the federation. They also use ontologies to classify the registries and to locate resources. This work does not deal with spatial modeling of OGC catalogs, so that spatial queries are not fully accomplished.

Boucelma et al. [Boucelma et al. 2002] integrate data and query language on heterogeneous spatial data sources distributed in a unique view. Their proposal is based on mediators and wrappers which access distributed data and implement query mapping. They use the global-as-view model [Chawathe et al. 1994], in which the wrappers enable not only answering queries on distributed data but also querying local data. The wrappers are built using components which interact with Web Feature Service. However, this system is based on a centralized architecture. Also, the inclusion of new services must be done manually.

Hubner et al. [Hubner et al. 2004] propose to unify distributed data sources in a unique point of interaction with the client, through a map definition tool. Their system uses metadata for resource location, which is classified using ontologies. Hence, semantic and syntactical searches may be achieved even with spatial data. However, the proposed architecture demands great effort from users. Moreover, data locating service is not fully compatible with OGC which penalizes interoperability. Also, the approach maintains a centered catalog.

The main contribution of our work is concerning the design of a SOA-based SDI which aims to integrate, locate, and catalog distributed spatial data sources (services and data). Mainly, we propose a catalog federation which enables dynamic exchange of heterogeneous SDIs. Hence, users have an integrated and transparent view of the available resources which makes easy data exchange and minimizes data replication. Moreover, by using Web services we provide interoperability, portability, flexibility and dynamic chaining, which are not present in geoportals and clearinghouses. Finally, the WS-GIS architecture combines GI Service composition with a service catalog federation, based on the ontologies, which enables a semantic modeling of catalogs.

Furthermore, the proposed communication protocol is spatial-aware. Hence, it is possible to rewrite queries on the fly so that only reminding data are forwarded to other catalogs in the federation. This spatial query optimization reduces data redundancy in the result set. To the best of our knowledge this approach has not been proposed yet in the specialized literature.

5. Conclusion

The large availability of spatial datasets and the advent of service-oriented architectures have motivated research on SDI based on Web services. In this paper we presented the WS-GIS architecture which aims to implement a SOA-based SDI which promotes a set of spatial services and a federation of catalogs to enable distributed queries. The main purpose of such architecture is to enable integration of heterogeneous spatial data sets, by promoting interoperability, flexibility, service chaining and scalability. We have also built a prototype to work in a Regional SDI, in a real scenario of managing water resources in the Brazilian State of Paraíba.

As further work we intend to improve the workflow service so that it can take into account load balancing, quality of service, data provenance and service context. Moreover, we intend to implement other spatial services such as OGC WCS; to migrate our Gazetteer Web service to be OGC compliant; and to analyze the use of grid computing in our architecture. Finally, it is important to address a study on usability in WS-GIS aiming for collecting the impacts of this infrastructure on e-Government.

References

Alameh, N. (2003) “Chaining Geographic Information Web Services”, In: IEEE Internet Computing, p. 22-29.

Almeida, D. R., Baptista, C. S., Silva, E. R., Campelo, C. E. C., Figueiredo, H. F. and Lacerda Y. A. (2006). A Context-Aware System Based on Service-Oriented Architecture. In: 20th International Conference on Advanced Information Networking and Applications, pages 205-210, IEEE Computer Society.

Alonso, G., Casati, F., Kuno, H. and Machiraju, V. (2004) Web Services: Concepts, Architectures and Applications, Springer-Verlag.

Baptista, C. S., Silva, E. R., Leite Jr., F. L. and Paiva, A. C. (2004) Using Open Source GIS in e-Government Applications. In International Conference on Eletronic Government, pages 418-421. Lecture Notes in Computer Science.

Bernard, L. and Craglia, M. (2005) SDI - From Spatial Data Infrastructure to Service Driven Infrastructure. In: *First Research Workshop on Cross-learning on Spatial Data Infrastructures and Information Infrastructures*, Enschede, Netherlands.

Boucelma, O., Essid, M. and Lacroix, Z.(2002) A WFS-based mediation system for GIS interoperability, In: *10th ACM International Symposium on Advances in Geographic Information Systems*, pages 23-28. ACM Press.

Burstein, M., Bussler, C., Zaremba, M., Finin, T., Huhns, M. N., Paolucci, M., Sheth, A. P. and Williams, S. (2005) “Semantic web services architecture”, IEEE Internet Computing, p. 72-81.

Chawathe, S., Garcia-Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y., Ullman, J.D. and Widom, J. (1994). The TSIMMIS Project: Integration of heterogeneous information sources, In: *16th Meeting of the Information Processing Society of Japan*, pages 7-18.

Crompvoets, J., Bregt, A., Rajabifard, A. and Williamson, I. (2004) “Assessing the worldwide developments of national spatial data clearinghouses”, In: International Journal of Geographical Information Science, p. 665-689.

Davis, C. and Alves, L. L. (2005) Local Spatial Data Infrastructures Based on a Service-Oriented Architecture, In: *Brazilian Symposium on GeoInformatics*, SBC.

Dijkstra, E. W. (1959) “A note on two problems in connection with graphs”, Numerische Mathematik, p. 269-271.

Fu, G., Jones, C. B. and Abdelmoty, A. I. (2005) Building a Geographical Ontology for Intelligent Spatial Search on the Web, In: *Databases and Applications*, pages 167-172.

Garcia-Molina, H., Ullman, J. D. and Widom, J. D. (2001), Database Systems: The Complete Book, Prentice Hall.

Georgiadou, P.Y. (2006) “SDI ontology and implications for research in the developing world”, International journal of spatial data infrastructures research, p. 51-64.

Gómez-Perez, A., Fernández-López, M. and Corcho, O. (2004), Ontological Engineering: with example from the areas of Knowledge Management, e-commerce and the Semantic Web, Springer-Verlag.

Hameed, A., Preece, A. and Sleeman, D. (2004). “Ontology reconciliations”, In: Handbook on Ontologies, Edited by S. Staab and R. Studer, Springer-Verlag, Germany.

Hübner, S., Spittel, R., Visser, U. and Vögele, T.J. (2004) “Ontology-Based Search for Interactive Digital Maps”, In: IEEE Intelligent Systems, p. 80-86.

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

“Open Geospatial Consortium”, Inc. http://www.opengeospatial.org, January 2006.

Pazinatto, E., Baptista, C. S. and Miranda, R. A. V. (2002). GeoLocalizador: um Sistema de Referência Espaço-Temporal Indireta utilizando um SGBD Objeto-Relacional. In: Brazilian Symposium on GeoInformatics, SBC.

Sivashanmugam, L., Verma, K. and Sheth, A. (2004) Discovery of Web Services in a Federated Registry Environment, In: *IEEE International Conference on Web Services* pages 270+. IEEE Computer Society.

Zhao, P., Chen, A., Liu, Y., Di, L., Yang, W. and Li, P. (2004) Grid metadata catalogservice-based OGC Web registry service, In: *12th Annual ACM International Workshop on Geographic Information Systems*, pages 22-30. ACM Press.