Quality Assessment
for Geographic Web Services

Pedro Ferreira Fernandes e Melo Medeiros

Dissertation for the achievement of the degree:
Master in Computer Engineering

Jury
President: Prof. José Tribolet
Supervisors: Prof. José Borbinha and Prof. Bruno Martins
External examiner: Prof. Mário Gomes

October 2009
Resumo

A capacidade de avaliar a qualidade de um serviço é um factor essencial para a diferenciação do sucesso dos fornecedores de serviços. A qualidade dos serviços Web geográficos pode ser medida através da avaliação de parâmetros standard de qualidade de serviço (e.g. disponibilidade, performance ou escalabilidade). Adicionalmente, parâmetros específicos do domínio devem também ser considerados (e.g. resolução ou precisão posicional dos dados geoespaciais). Nesta perspectiva, a presente dissertação propõe uma Framework para avaliar a qualidade de serviços Web geográficos, assim como a qualidade dos dados que estes fornecem. A Framework proposta foi implementada no sistema GeoWatchDog (GWD). O GWD foi avaliado usando serviços disponibilizados pelo Instituto Geográfico Português (IGP) e pelo portal Espanhol IDEE. Os resultados destas avaliações são também apresentados nesta dissertação.

**Palavras-chave:** SIG, Serviços Web Geográficos, Qualidade de Serviço, Qualidade de Dados
Abstract

Being able to assess the quality of a service is a significant factor in distinguishing the success of service providers. In this context, the quality of geographic Web services can be measured through the assessment of standard Quality of Service (QoS) parameters (e.g. availability or scalability) plus of other non-standard parameters specific of the domain (e.g. resolution or positional accuracy of the geospatial data). From this perspective, this dissertation proposes a framework to assess the quality of geographic Web services, as well as the quality of the data that they provide. The proposed framework was implemented in the GeoWatchDog (GWD) system. The GWD was evaluated using services available from the Portuguese Geographic Institute (IGP) and the Spanish IDEE portal. The results of these evaluations are also presented in this dissertation. Finally, the GWD is all implemented in Java, and will be released as open-source.

**Keywords:** GIS, Geographic Web Services, Quality of Service, Data Quality
## Contents

### List of Figures

vi

### List of Tables

viii

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Problem</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Solution</td>
<td>1</td>
</tr>
<tr>
<td>1.3 Contributions</td>
<td>2</td>
</tr>
<tr>
<td>1.4 Outline</td>
<td>2</td>
</tr>
<tr>
<td>2 Related Work</td>
<td>3</td>
</tr>
<tr>
<td>2.1 Geographic Information Systems on the Web</td>
<td>3</td>
</tr>
<tr>
<td>2.2 Web Services Technologies</td>
<td>3</td>
</tr>
<tr>
<td>2.3 OGC Web Service Standards</td>
<td>4</td>
</tr>
<tr>
<td>2.3.1 Representation of Geographic Information</td>
<td>4</td>
</tr>
<tr>
<td>2.3.1.1 OpenGIS Geography Markup Language Specification</td>
<td>5</td>
</tr>
<tr>
<td>2.3.1.2 OpenGIS Keyhole Markup Language Specification</td>
<td>6</td>
</tr>
<tr>
<td>2.3.1.3 Comparison between GML and KML</td>
<td>6</td>
</tr>
<tr>
<td>2.3.2 OGC Geographic Web Services</td>
<td>7</td>
</tr>
<tr>
<td>2.3.2.1 OpenGIS Web Feature Service Specification</td>
<td>7</td>
</tr>
<tr>
<td>2.3.2.2 OpenGIS Web Map Service Specification</td>
<td>8</td>
</tr>
<tr>
<td>2.3.2.3 OpenGIS Web Coverage Service</td>
<td>9</td>
</tr>
<tr>
<td>2.3.2.4 OGC Gazetteer Service Specification</td>
<td>9</td>
</tr>
<tr>
<td>2.3.2.5 OpenGIS Web Processing Service Specification</td>
<td>10</td>
</tr>
<tr>
<td>2.3.2.6 OpenGIS Catalogue Services Web</td>
<td>10</td>
</tr>
<tr>
<td>2.3.3 Implementations of OGC Standards</td>
<td>11</td>
</tr>
<tr>
<td>2.4 INSPIRE</td>
<td>12</td>
</tr>
<tr>
<td>2.5 Quality of Service</td>
<td>12</td>
</tr>
<tr>
<td>2.5.1 Quality of Service Dimensions</td>
<td>12</td>
</tr>
<tr>
<td>2.5.2 INSPIRE Quality of Service Guidelines</td>
<td>14</td>
</tr>
</tbody>
</table>
### 2.5.3 Assessing the Quality of OGC Web Services - Deegree owsWatch

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5.3 Assessing the Quality of OGC Web Services - Deegree owsWatch</td>
<td>15</td>
</tr>
</tbody>
</table>

### 2.6 Geospatial Data Quality

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.6.1 Geospatial Data Quality Dimensions</td>
<td>15</td>
</tr>
<tr>
<td>2.6.2 Assessing Geospatial Data Quality</td>
<td>15</td>
</tr>
<tr>
<td>2.6.2.1 Point-based Assessment</td>
<td>16</td>
</tr>
<tr>
<td>2.6.2.2 Line-based Assessment</td>
<td>16</td>
</tr>
</tbody>
</table>

### 3 Requirements for Assessing the Quality of Geographic Web Services

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Administrating the System</td>
<td>20</td>
</tr>
<tr>
<td>3.2 Monitoring the Performance and Availability of Geographic Web Services</td>
<td>20</td>
</tr>
<tr>
<td>3.3 Testing the Scalability of Geographic Web Services</td>
<td>21</td>
</tr>
<tr>
<td>3.4 Assessing Geospatial Data Quality</td>
<td>21</td>
</tr>
<tr>
<td>3.5 Testing Geographic Web Services</td>
<td>22</td>
</tr>
</tbody>
</table>

### 4 The GeoWatchDog Application

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 GWD Model</td>
<td>25</td>
</tr>
<tr>
<td>4.1.1 Configuration File</td>
<td>26</td>
</tr>
<tr>
<td>4.1.2 Services File</td>
<td>26</td>
</tr>
<tr>
<td>4.1.3 Request Files</td>
<td>28</td>
</tr>
<tr>
<td>4.1.4 Test-Plan Files</td>
<td>28</td>
</tr>
<tr>
<td>4.1.5 Service URL files</td>
<td>29</td>
</tr>
<tr>
<td>4.2 GWD Controller</td>
<td>29</td>
</tr>
<tr>
<td>4.2.1 GWD Administrator Component</td>
<td>29</td>
</tr>
<tr>
<td>4.2.2 GWD Monitor Component</td>
<td>30</td>
</tr>
<tr>
<td>4.2.3 GWD Load-Tester Component</td>
<td>33</td>
</tr>
<tr>
<td>4.2.4 GWD Data-Tester Component</td>
<td>33</td>
</tr>
<tr>
<td>4.2.4.1 Assessing Geographic Coordinates Resolution</td>
<td>33</td>
</tr>
<tr>
<td>4.2.4.2 Assessing Map Images Resolution</td>
<td>36</td>
</tr>
<tr>
<td>4.2.5 GWD Playground Component</td>
<td>37</td>
</tr>
<tr>
<td>4.3 GWD View</td>
<td>37</td>
</tr>
<tr>
<td>4.3.1 GWD Administrator Interface</td>
<td>40</td>
</tr>
<tr>
<td>4.3.2 GWD Monitor Interface</td>
<td>43</td>
</tr>
<tr>
<td>4.3.3 GWD Load-Tester Interface</td>
<td>48</td>
</tr>
<tr>
<td>4.3.4 GWD Data-Tester Interface</td>
<td>50</td>
</tr>
<tr>
<td>4.3.5 GWD Playground Interface</td>
<td>52</td>
</tr>
</tbody>
</table>

### 5 Evaluation

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Registration of Services published by a Catalogue</td>
<td>56</td>
</tr>
<tr>
<td>5.2 Evaluation of the Services Monitoring</td>
<td>58</td>
</tr>
<tr>
<td>5.3 Evaluation of the Services Load-Testing</td>
<td>59</td>
</tr>
<tr>
<td>5.4 Evaluation of the Data Quality Assessment</td>
<td>62</td>
</tr>
</tbody>
</table>
## List of Figures

2.1 Sample of a GML code. .................................................. 5  
2.2 Sample of a KML code. .................................................. 6  
2.3 Sample of a “GetFeature” request. ................................. 7  
2.4 KVP for a “GetFeature” request. ................................. 7  
2.5 “GetFeature” Response. ............................................... 8  
3.1 Use cases ................................................................. 19  
4.1 GWD components diagram ........................................... 25  
4.2 Sample of the configuration file. .................................. 26  
4.3 Services file ........................................................... 27  
4.4 GWD requests directory .............................................. 28  
4.5 XML request ........................................................... 28  
4.6 Service URL file ...................................................... 29  
4.7 Activity diagram of the GWD Administrator component ....... 31  
4.8 Activity diagram of the GWD Monitor component .............. 32  
4.9 Activity diagram of the GWD Load-Tester component ......... 34  
4.10 Activity diagram of the GWD Data-Tester component ......... 35  
4.11 Activity diagram of the GWD Playground component ......... 38  
4.12 Services list view ..................................................... 39  
4.13 Service details view .................................................. 39  
4.14 Request details ........................................................ 40  
4.15 HTML snippet form .................................................. 41  
4.16 Interface of the GWD Administrator component .............. 41  
4.17 Interface of the GWD Administrator component to register a service .................................................. 42  
4.18 Catalogue client interface .......................................... 43  
4.19 Services returned by a catalogue .................................. 44  
4.20 Monitoring list ......................................................... 45  
4.21 Configure the monitoring function ................................ 46  
4.22 GWD Monitor results ................................................ 47
4.23 GWD Load-Tester services list .................................................. 48
4.24 GWD Load-Tester test-plan ......................................................... 49
4.25 GWD Load-Tester test-plan .......................................................... 50
4.26 Algorithms list ............................................................................. 50
4.27 Service client interface of the GWD Data-Tester component .......... 50
4.28 Results of the GWD Data-Tester component ............................... 51
4.29 Interface of the GWD Playground component ............................... 53

5.1 “GetFeature” request ..................................................................... 55
5.2 “GazetterQuery” request ................................................................. 56
5.3 Query sent to the SNIG catalogue ................................................... 57
5.4 Performance graph of the “SNIG CAOP Continente” ..................... 59
5.5 Performance graph of the Digmap ................................................. 60
5.6 Response returned by the service “SNIG CAOP Continente” .......... 66
5.7 Response returned by the service “MassGIS WMS” ....................... 66
5.8 Response returned by the service “MassGIS WMS” ....................... 67

6.1 Algorithms file ............................................................................... 71
# List of Tables

2.1 OGC Implementations ......................................................... 11  
2.2 OGC standards ensuring the network services requirements. .............. 12  

5.1 Services registered in the GWD application. ............................ 54  
5.2 Parameters of the “GetMap” requests .................................... 55  
5.3 Parameters of the “GetCoverage” request ................................ 56  
5.4 SNIG services ....................................................................... 57  
5.5 Performance and availability values ........................................ 61  
5.6 Parameters of the “GetMap” requests .................................... 61  
5.7 Load-testing results. ............................................................. 62  
5.8 Results of the coordinates precision algorithm. .......................... 63  
5.9 WMS tested. ................................................................. 64  
5.10 Parameters of the “GetMap” request for the service “CAOP Acores” .... 64  
5.11 Results of the map image resolution algorithm .......................... 64  
5.12 Parameters of the “GetMap” requests .................................... 65  
5.13 Validation matrix ............................................................... 68
Chapter 1

Introduction

Enterprises in the public and private sectors have recently produced a surge in Web services and Web applications for geographic information systems (GIS), making large spatial-data archives available over the Internet. Technically, Web services technologies have provided the necessary standards for applications to integrate with GIS data and services. Furthermore, the Open Geospatial Consortium (OGC) developed standards for interoperable geospatial Web services that can be published, discovered and invoked across the Web. Implementing these services and enabling them to integrate with non-spatial Web services will help bring the value of geospatial applications to a much broader community.

1.1 Problem

Although the importance of geographic Web services is well-established, their quality is often questionable. Moreover, the issue of geospatial data quality has an important role for the adoption of certain geographic Web services. From this perspective, with the widespread availability of geographic Web services, their associated quality of service will become a significant factor in distinguishing the success of service providers. In this context, the quality of geographic Web services can be measured through the assessment of standard Quality of Service (QoS) parameters (e.g. availability or scalability), plus of other non-standard parameters specific of the domain (e.g. resolution or positional accuracy of the spatial data).

1.2 Solution

Being able to assess the quality of a service is a significant factor in distinguishing the success of service providers. This dissertation proposes a framework to assess the quality of geographic
Web services, as well as the quality of the data that they provide.

The proposed framework allows a human operator to assess the quality of geographic Web services through three main functions: a) monitoring of the service performance and availability, b) test service scalability, c) assessment of the geospatial data quality returned by the services. The Web services that can be assessed by this framework are expected to implement open standards such as those provided by the OGC.

The proposed framework was implemented in the GeoWatchDog (GWD) system. The GWD application has three components, one for each assessment function, plus a specific component for administration operations and another for testing operations. The GWD Administrator component manages the services registered in the GWD application. The GWD Monitor component monitors the services performance and availability, by periodically sending pre-defined requests to the services. The GWD Load-Tester component executes load tests in order to assess scalability. The GWD Data-Tester component applies data quality algorithms to the data returned by the services. Finally, the GWD Playground component is an operational module from where a user can invoke and assess the responses returned by the services.

The GWD was evaluated using services available from the a) Portuguese Geographic Institute (IGP), b) Spanish IDEE portal and c) other services available on the Web.

1.3 Contributions

This work proved that the quality of geographic Web services can be analysed from three different perspectives, namely 1) performance and availability monitoring, 2) scalability testing, 3) data quality assurance. A software framework capable of supporting all the perspectives was designed and evaluated with services from two national providers.

1.4 Outline

The rest of this dissertation is organized as follows. Chapter 2 describes the background concepts and related work that constitutes the basis for this work. Chapter 3 identifies the requirements of the proposed framework. Chapter 4 describes the implementation of the framework in the GWD application. Chapter 5 describes the evaluation of the GWD application and presents the results. Finally, Chapter 6 resumes the main ideas and conclusions and discusses possible directions for future work.
Chapter 2

Related Work

This chapter presents the background and the related work that constitute the basis for the research presented in this dissertation.

2.1 Geographic Information Systems on the Web

A geographic information system (GIS) can be defined as a special type of information system tailored to store, process, and manipulate geospatial data [1]. A GIS gathers a diversity of data, spatial or not, from different source, providing services to manipulate and visualize geographic information. The Web services techniques and its associated technologies led the GIS to the Web. The GIS behavior results from the interoperability between a set of Web services for dealing and manipulating geospatial data, such as, map servers, geospatial databases, gazetteers, etc.

2.2 Web Services Technologies

A Web service is a software system that supports interaction between different machines over the Web [2]. Normally, a Web service has an application programming interface (API) available on the network in order to allow the remote execution of service operations. The Web service is invoked through HTTP and data is usually exchanged in XML formats.

Currently, Web services can be categorized as “big Web services” and “RESTful Web services”. The big Web services use XML messages which implement the Simple Object Access Protocol (SOAP) [3] and, normally, use the Web Service Description Language (WSDL) [4] for describing contractually the operations implemented by the service. The RESTful Web services are more simple. They do not implement SOAP, neither use the WSDL for describing the API.
The Web services can be built on service oriented architectures (SOA) [5]. The service requesters can discover the providers through mechanisms used to publish and discover the collections of available services. For example, the Universal Description, Discovery and Integration (UDDI) [6]. The UDDI represents a group of Web-based registries that expose information about Web services and their technical interfaces (or API).

2.3 OGC Web Service Standards

The W3C purposed the SOA Web service architecture based on a trio of standards — SOAP, WSDL, UDDI —, often coupled with others for business processes, security, coordination, transaction, etc. In parallel with the development of these general-purpose Web services standards, the Open Geospatial Consortium\(^1\) (OGC) encouraged the development and implementation of standards for processing and sharing of geospatial content [7]. The OGC Web Service (OWS) standards were developed to create self-contained, standards-based, interoperable geospatial web services that can be published, discovered and invoked across the web. Some of the OWS standards, described in the following sub-sections, support application developers in integrating a variety of online geoprocessing and location services. Implementing these services and enabling them to integrate with non-spatial Web services will help bring the value of geospatial applications to a much broader community.

Conceptually, the OWS architecture includes standards for service discovery, description, and binding layers corresponding to UDDI, WSDL, and SOAP in the W3C architecture. Rather than general issues, however, the OGC intended to specify only those issues that are specific to geographic information. Currently, OGC Web services are not equivalent to the W3C SOAP-based Web services, although the OGC is attempting to integrate the Web services standards into the OWS framework, through changes in the common OWS architecture and by providing WSDL descriptions for their service API.

2.3.1 Representation of Geographic Information

Geographic information derives from two categories of geographic phenomena — discrete and continuous. Discrete phenomena are recognizable objects that have relatively well-defined boundaries and are associated with a location relative to the Earth. Continuous phenomena vary over space and have no specific extent. A value or description of a continuous phenomenon is only meaningful at a particular position in space (and possibly time).

In the context of the OGC Abstract Specification [8], we have the concept of "geographic features" for representing discrete phenomena. Features [9] are abstractions of real world phenomena (ISO

\(^1\)http://www.opengeospatial.org
19101:2002), represented by vector data. The spatial characteristics described by each feature are represented by sets of geometric primitives (points, curves, surfaces or solids). Other characteristics of the phenomenon are recorded as feature attributes. The ISO 19107:2003 standard defines a schema for describing features in terms of geometric and topological primitives.

For representing continuous phenomena, the OGC Abstract Specification uses the term “coverage” to refer to any data representation that assigns values directly to spatial position. A coverage [10] is a function from a spatial, temporal or spatiotemporal domain to an attribute range. More specifically, coverage is a subtype of feature that has multiple values for each attribute type.

2.3.1.1 OpenGIS Geography Markup Language Specification

The OpenGIS Geography Markup Language [11](GML) is used for encoding geographic information and sharing it between Web services. This language describes features with geographic properties. Therefore, a GML document specifies data sets that contain points, lines and polygons, the principal geometric objects in GML. The last version of this specification (GML 3.0), temporal information can also be described.

GML is based in XML schemas that define its structure. Two of these schemas are the Geometry Schema, which defines allowed geometries (Point, LineString, LinearRing, Polygon, etc.) and the Feature Schema, which defines the geographic properties for features.

In brief, a feature specified with GML is a XML element. The name of the feature specifies its type. The content of the feature defines the elements that describe the feature as a properties set. Figure 2.1 presents a GML document that describes one feature corresponding to a school.

```
<Feature fid="142" featureType="school" Description="A middle school">
  <Polygon name="extent" srsName="epsg:27354">
    <LineString name="extent" srsName="epsg:27354">
      <CData>
        491888.999999459,5458045.99963358
        491904.999999458,5458044.99963358
        491908.999999462,5458064.99963358
        491924.999999461,5458064.99963358
        491925.999999462,5458079.99963359
        491977.999999466,5458120.99963360
        491953.999999466,5458017.99963357
      </CData>
    </LineString>
  </Polygon>
</Feature>
```

**Figure 2.1:** Sample of a GML code.
2.3.1.2 OpenGIS Keyhole Markup Language Specification

The Keyhole Markup Language [12] (KML) was initially developed by Google to describe and present geographic information on Google Earth. Nowadays, like GML, KML is also an OGC standard that can be used to share geographic information between web services.

As a XML based language, KML has a structure with a nested set of elements to describe geographic features and the way those are presented in systems such as Google Earth and Microsoft Virtual Earth.

To describe a feature with KML, we use feature elements such as a Placemark (an element with a geometrical description), Overlay (an area overlaid on the ground or the screen), or a Region (a geographic space that can be used to trigger events). Together, these features provide flexibility in the specification of interactive geographic applications. Figure 2.2 presents a KML document that describes one feature with a placemark and an associated polygon.

```xml
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Placemark>
    <name>A middle school</name>
    <Polygon>
      <extrude>1</extrude>
      <altitudeMode>relativeToGround</altitudeMode>
      <outerBoundaryIs>
        <LinearRing>
          <coordinates>
            491888.999999459,5458045.99963358
            491904.999999458,5458044.99963358
            491908.999999462,5458064.99963358
            491924.999999461,5458064.99963358
            491925.999999462,5458079.99963359
            491977.999999466,5458120.9996336
            491953.999999466,5458017.99963357
          </coordinates>
        </LinearRing>
      </outerBoundaryIs>
    </Polygon>
  </Placemark>
</kml>
```

Figure 2.2: Sample of a KML code.

2.3.1.3 Comparison between GML and KML

Although GML and KML have similarities in their structure (KML came afterwards and was inspired by GML) and both are intended to facilitate the share of geographic information, they have distinct objectives. While GML is focused on description of geographic features, KML is focused on geographic visualization. Geographic visualization includes not only the presentation of graphical data, but also the control of the user’s navigation in the sense of where to go and
where to look. From this perspective, KML is complementary to most of the key existing OGC standards, including GML, WFS (Web Feature Service) and WMS (Web Map Service).

2.3.2 OGC Geographic Web Services

The OGC Web Service (OWS) specifications are build on standard representation schemes such as those referred in previously sub-section. Ideally, each OWS define two methods of encoding its requests. The first uses XML as the encoding language, and it is intended to be used with HTTP post method. The second encoding uses keyword-value pairs (KVP) to encode the various parameters of a request and it is intended to be used with HTTP get method.

As an example of a XML request that could be sent to a OWS, Figure 2.3 presents a request to obtain features, from a OGC Web Feature Service, having a “typename” equal to “vg”. Figure 2.4 presents an equivalent request, using the KVP encoding. Figure 2.5 presents an excerpt of the XML response for the requests referred above. More sample requests for the OGC specifications, can be found in their specification documents provided by the OGC in its site.

Besides the specific operations of each service, all OWS have a “GetCapabilities” operation, which returns service capabilities.

```xml
<wfs:GetFeature xmlns:wfs="http://www.opengis.net/wfs"
    xmlns:ogc="http://www.opengis.net/ogc"
    xmlns:gml="http://www.opengis.net/gml"
    xmlns:cgi="http://www.opengis.net/cite/geometry"
    version="1.0.0" service="WFS" outputFormat=GML2>
    <wfs:Query typeName="vg"/>
</wfs:GetFeature>
```

**Figure 2.3**: Sample of a “GetFeature” request.

```text
http://mapas.igeo.pt/ows/vgs?version=1.0.0&request=getfeature&typename=vg&service=wfs
```

**Figure 2.4**: KVP for a “GetFeature” request.

2.3.2.1 OpenGIS Web Feature Service Specification

Languages such as KML and GML describe features. In order to access these features we have the OGC Web Feature Service (WFS, ISO/CD 19142) Implementation Specification [13].

The WFS standard defines an interface for access and manipulation of geographic features. Examples of these operations are deleting, creating and updating features, as well as querying features based on spatial and non-spatial constraints. In other words, a WFS is a wrapper over a feature database with a web service interface that allows access and manipulation of features.

The WFS interface defines the following specific operations:

- **DescribeFeatureType** – returns the structure of any allowed feature type;
- **GetFeature** – returns feature instances according to the client specifications such as, which feature properties to fetch or query constraints.

### 2.3.2.2 OpenGIS Web Map Service Specification

To present the features over a map we have the OGC Web Map Service (WMS, ISO 19128) Implementation Specification [14]. The WMS standard defines interfaces that give an interoperable way to combine and view maps images (PNG, GIF or JPEG) in which geographic features can be rendered.

The OpenGIS Styled Layer Descriptor [15] (SLD) defines a mechanism for user-defined symbolization of feature data in a WMS. In brief, an SLD-enabled WMS retrieves feature data from a WFS and applies styling information provided by the user in order to render a map.

The SLD-enabled WMS interface defines the following specific operations:

- **GetMap** – returns a map satisfying the request parameters;
• GetFeatureInfo – returns information about the feature presented in maps;
• DescribeLayer – given the name of a layer, returns information about it (e.g. the features in the layer and their types);
• GetLegendGraphic – returns the symbolization (images) used to represent specific features in the map;
• GetStyles – allows the user to download a style;
• PutStyles – allows the user to upload a style.

2.3.2.3 OpenGIS Web Coverage Service

In order to support the retrieval of geospatial data as “coverages”, the Web Coverage Service (WCS) Implementation Specification [16] defines an interface that allows clients to request server’s information based on spatial constraints and other criteria.

Unlike the WMS, which portrays spatial data to return static maps (rendered as pictures by the server), the WCS provides the available data together with their detailed descriptions. It defines a rich syntax for requests against these data and supports analysis over the data. Unlike the WFS, which returns discrete geospatial features, the Web Coverage Service returns coverages representing space-varying phenomena, relating a spatio-temporal domains to a range of properties.

The WCS interface defines the following specific operations:

• DescribeCoverage – lets clients request a full description of one or more coverages served by a particular WCS server;
• GetCoverage – returns a coverage encoded in a well-known coverage format (e.g. GeoTIFF3, HDF-EOS4, NITF5 and CF-NetCDF6).

2.3.2.4 OGC Gazetteer Service Specification

A Gazetteer can be defined as a geospatial dictionary of geographic names. The entries in a gazetteer have a name, location (coordinates representing a point, line or areal location) and a type associated to one feature [17]. The OGC Gazetteer Service specification (WFS-G) [18] extends the WFS interface in order to support features as geographic names.

Since the WFS-G is a WFS extension, both services define the same operations (i.e. “DescribeFeatureType”, “GetFeature” and “GetCapabilities”). Additionally, the WFS-G interface

---

3http://www.remotesensing.org/geotiff/geotiff.html
4http://www.hdfeos.org
5http://www.ismc.nga.mil/ntb/baseline/1999.html
6http://www.cgd.ucar.edu/cms/eaton/cf-metadata
defines the “GetGMLObject” operation to retrieve element instances by traversing XLinks that refer to their XML identifiers.

The WFS-G is based on the gazetteer protocol project developed by the Alexandria Digital Library (ADL) - ADL Gazetteer Protocol [19]. As an example that implements the WFS-G, we have the DIGMAP Gazetteer [20].

2.3.2.5 OpenGIS Web Processing Service Specification

The OGC Web Processing Service (WPS) Implementation Specification [21] defines an interface that facilitates the publishing of geospatial processes, their discovery of and their execution by clients. Normally, these processes are algorithms and calculations on geospatial data. The WPS interface also specifies the way clients request the execution of processes and descriptive information about its.

The WPS interface defines the following specific operations:

- DescribeProcess – allows WPS clients to request a full description of one or more processes that can be executed by the “Execute” operation. This description includes the input and output parameters and formats;
- Execute – allows WPS clients to run a specified process implemented by a server.

2.3.2.6 OpenGIS Catalogue Services Web

With the increasing availability of spatial data on the Web, the role of catalogue services has gained greater importance. These services act as a broker between data providers and data requesters, providing a convenient way for the service discovery. Clients can search for Web services that match specific criteria or browse the registered web services stored in the catalogue’s repository [22]. The OGC Catalogue Services Implementation Specification [23] specifies the interfaces, bindings, and a framework for defining application profiles required to publish and access digital catalogues of metadata for geospatial data, services, and related resource information. Metadata acts as generalized properties that can be queried and returned through catalogue services for resource evaluation and, in many cases, invocation or retrieval of the referenced resource.

In order to describe the request and response messages that are common to all Web-based catalogue services, the OGC proposes the Catalogue Services Web interface (CSW, ISO 19115/19119). The CSW ensures the interaction between a client and a server using a standard request-response model of the HTTP protocol. Request and response messages are encoded as KVP within a request URI or using an XML entity-body. Requests may also be embedded in a messaging framework such as SOAP.
The CSW interface defines the following specific operations:

- DescribeRecord – allows a client to discover elements of the information model supported by the target catalogue service;
- GetDomain – used to obtain runtime information about the range of values of a metadata record element or request parameter;
- GetRecords – allows the search for records that match specific criteria;
- GetRecordsById – retrieves the default representation of catalogue records using their identifier.

The CSW does not support queries by service type. However, in [24], additional requirements are proposed to extend CSW in order to enable users to query by service type.

2.3.3 Implementations of OGC Standards

There are several open-source products implementing the OGC specifications that were previously discussed.

The deegree project is one of the most extensive implementation of OGC standards in the field of Free Software[25]. The deegree is a Java framework offering the main building blocks for Spatial Data Infrastructures (SDI). Its entire architecture is developed using standards of the OGC and ISO/TC 211 (ISO Technical Committee 211 – Geographic Information/Geomatics). The deegree framework encompasses OGC Web services as well as clients and security components. An overview of all components implemented by deegree can be found at lat/lon’s site.

Table 2.1 lists the most well-known implementations of OGC standards. Analyzing this table we can conclude the following:

- deegree implements all the considered OGC specifications, except the KML;
- GeoServer implements all the considered OGC specifications, except the WPS;
- WPS is only implemented by the deegree;
- The GML, WFS and WMS are implemented by all the considered products.

<table>
<thead>
<tr>
<th>Product</th>
<th>GML</th>
<th>KML</th>
<th>WFS</th>
<th>WMS</th>
<th>WPS</th>
<th>CSW</th>
<th>WCS</th>
<th>GAZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>deegree</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MapServer</td>
<td>✓</td>
<td>_</td>
<td>✓</td>
<td>✓</td>
<td>_</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>GeoServer</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>_</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>OpenLayers</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
</tbody>
</table>

Table 2.1: Implementations of the OGC specifications.

---

7http://www.latlon.de


2.4 INSPIRE

In May 2007 the INSPIRE [26] directive entered in force. Its objective was to establish an infrastructure for spatial information in Europe to support Community environmental policies, and policies or activities which may have an impact on the environment. The INSPIRE directive was a main development in terms of SDI.

The INSPIRE directive proposed Network Services for sharing spatial data between the various levels of public authority in the Community. Those network services should make it possible to discover, transform, view and download spatial data and to invoke spatial data and e-commerce services.

There are OGC standards that ensure the requirements of the INSPIRE directive for the network services. Table 2.2 shows the standards which ensures the identified requirements.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>CSW</th>
<th>WMS</th>
<th>WFS</th>
<th>WCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discover</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>View</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Download</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 2.2: OGC standards ensuring the network services requirements.

2.5 Quality of Service

Quality of service (QoS) refers to non-functional properties of Web services such as performance, reliability, availability, security, etc. INSPIRE directive established QoS criteria which shall be ensured by geographic Web services. The next two sub-sections present the standard QoS dimensions and the INSPIRE QoS guidelines. Finally, the third sub-section presents the owsWatch, a Web-based tool to monitor the performance and availability of geographic Web services.

2.5.1 Quality of Service Dimensions

There is a wealth of proposals for QoS dimensions, coming from different sources, such as, the W3C standards organization, the IBM organization and a set of articles. The W3C, in [27], proposes quality aspect of a web service to include performance, reliability, scalability, capacity, robustness, exception handling, accuracy, integrity, accessibility, availability, interoperability, accuracy, integrity, accessibility, availability, interoperability,

---

8 http://www.deegree.org
9 http://mapserver.gis.umn.edu
10 http://geoserver.org
11 http://openlayers.org
security, and network-related QoS requirements. These quality attributes are defined as follows:

- **Performance** – represents how fast a service request can be completed;
- **Reliability** – represents the ability of a web service to perform its required functions under stated conditions for a specified time interval. The reliability is the overall measure of a web service to maintain its service quality;
- **Scalability** – represents the capability of increasing the computing capacity of service provider’s computer system and system’s ability to process more users’ requests, operations or transactions in a given time interval;
- **Capacity** – is the limit of the number of simultaneous requests which should be provided with guaranteed performance;
- **Robustness** – represents the degree to which a web service can function correctly even in the presence of invalid, incomplete or conflicting inputs;
- **Exception Handling** – Web services should be provided with the functionality of exception handling. Since it is not possible for the service designer to specify all the possible outcomes and alternatives (especially with various special cases and unanticipated possibilities), exceptions should be handled properly;
- **Accuracy** – is defined as the error rate generated by the web service;
- **Integrity** – integrity for web services should be provided so that a system or component can prevent unauthorized access to, or modification of, computer programs or data;
- **Accessibility** – represents whether the web service is capable of serving the client’s request;
- **Availability** – is the probability that the system is up;
- **Interoperability** – Web services should be interoperable between the different development environments used to implement services;
- **Security** – Web services should be provided with the required security.

There are differences between the above definitions and other proposals. For example, according to Mani and Nagarajan [28], two IBM software engineers, scalability and performance of Web services can be defined as follows:

- **Scalability** – ability to consistently serve the requests despite variations in the volume of requests;
- **Performance** – measured in terms of throughput and latency. Throughput represents the number of Web service requests served at a given time period. Latency is the round-trip time between sending a request and receiving the response.
The QoS dimensions presented in the previous sub-section are suitable for the general Web services. Based on these dimensions, the INSPIRE directive proposes QoS guidelines for its network services. The following sub-section presents those guidelines.

2.5.2 INSPIRE Quality of Service Guidelines

In the process of putting forward guidelines, the applicability to the INSPIRE context of the W3C QoS dimensions is first assessed, and then commonalities with the other sources are identified to finally form the base of the proposed guidelines [29]. Therefore, the QoS dimensions proposed to guide the definition of minimum performance criteria for the INSPIRE Network Services are the following:

- Performance;
- Reliability;
- Capacity;
- Availability;
- Security;
- Regulatory;
- Interoperability.

The INSPIRE defined requirements for some of the criteria presented above. In terms of performance the requirements are as follows:

- The response time for sending the initial response to a discovery service request shall be maximum 3 seconds in normal situation;
- For a 470 Kilobytes image (e.g. 800x600 pixels with a color depth of 8 bits), the response time for sending the initial response to a Get Map Request to a view service shall be maximum 5 seconds in normal situation;
- Normal situation represents periods out of peak load. It is set at 90% of the time.

In terms of capacity the requirements are as follows:

- The minimum number of served simultaneous requests to a discovery service according to the performance Quality of Service shall be 30 per second;
- The minimum number of served simultaneous service requests to a view service according to the performance quality of service shall be 20 per second.

In terms of availability the requirements are as follows:

- The probability of a Network Service to be available shall be 99% of the time.
2.5.3 Assessing the Quality of OGC Web Services - Deegree owsWatch

In order to monitor the availability of different OGC web services, the project Deegree integrates a Web-based application, the owsWatch.

owsWatch notifies the user by email (or through its interface) if any of the monitored services is experiencing any problems. These might be high traffic that caused response latency or the service suffered a failure altogether. Monitoring a service is done by sending configurable requests (like GetMap for a WMS). These requests can be configured by the owsWatch user with certain constraints on that test (e.g., timeout, repeat times) to test if the service is running. The main characteristics of owsWatch are as follows:

- Supports HTTP GET (with KVP requests) and HTTP POST (with XML requests);
- Supports OGC web services in different versions: WMS 1.1.1, WFS 1.1.0, WCS 1.0.0, CSW 2.0.2, SOS 1.0.0;
- Supports GetCapabilities for all supported service types;
- Supports a service specific request for each service type (WMS:GetMap, WFS:GetFeature, WCS:GetCoverage, CSW:GetRecords, SOS:DescribeSensor);
- Notifies the user per mail on case of response failure;
- Generates protocol files for each monitored service instance.

Besides the monitoring function, the owsWatch also permits the users to test the services by sending standardized requests and visualizing the obtained responses through a service client interface.

2.6 Geospatial Data Quality

According to ISO standard 8402, quality is defined as “totality of characteristics of a product that bear on its ability to satisfy stated and implied needs”. In the context of geographic Web services, verifying the quality of geospatial data is essential to data sharing. The following two sub-sections present dimensions and techniques to assess geospatial data quality.

2.6.1 Geospatial Data Quality Dimensions

Some previous studies have proposed geospatial criteria that can be used alongside with the QoS dimensions in the discovery and invocation of geographic Web services [30]. The ISO 19113/19114 standards embody principles and evaluation procedures for geographic information
[31], and the main elements of geospatial data quality were recently summarized in [32]. Hence, 
the geospatial data quality dimensions can be summarized as:

- **Lineage** - description of the source material from which the data were derived, and the 
  methods of derivation, including all transformations involved in the production process;
- **Positional accuracy** - measures how well the true measurements of an object on the earth’s 
  surface match the same object stored as series of digital coordinates in geospatial dataset;
- **Attribute accuracy** - accuracy of all attributes other than the positional and temporal 
  attributes;
- **Logical consistency** - the fidelity of relationships (e.g. topological relationships) encoded 
  in the data;
- **Completeness** - a measure of the absence of data and the presence of excess data. Errors 
  resulting in over-completeness are called errors of commission, and errors resulting in 
  incompleteness are called errors of omission;
- **Temporal accuracy** - summary of errors in time measurements;
- **Variation in quality** - a textual and qualitative description of the expected or tested uni-
  formity of quality parameters in a geographic data set;
- **Meta-quality** - information on the quality of the quality description;
- **Resolution** - is the detail at which data are presented.

Besides the above dimensions, it is important to define what is an error in the context of geospa-

tial data. Error can be defined as the difference between the measured value of a property and 
the true value of the same property. The measurement of error therefore requires agreement on 
a clear definition of what is perceived as reality. Precision is a summary measure of error.

### 2.6.2 Assessing Geospatial Data Quality

Geospatial data are collected from various sources and associated together to derive valuable 
products. Hence, the consistency among their positions and attributes are the biggest concerns 
in applications utilizing data sets from multiple sources [33]. The quality assessment of such data 
generally can be seen to consist of three steps, a) check of logical consistency, b) verification and 
c) update. The logical consistency is ensured by comparing the data with the data model, i.e. 
the existence of attributes and the consistency of geometry etc. are checked. During verification 
and update the existing data are assessed using reference information. In the verification step 
the geometric accuracy and the correctness of attributes (if available in the reference) are assessed 
[34].
Most procedural checks require manual validation, but there are mechanisms designed to automatically verify the integrity of a GIS database [35]. As an example, the algorithms to assess the quality of linear data (e.g. lines representing roads, streets, boundaries, etc.) presented in the next two sub-sections. Other possible geospatial data quality measurements search for logical inconsistencies and missing or strange attribute values. For instance, attribute values must contain acceptable or default range values.

2.6.2.1 Point-based Assessment

The point-based algorithm [33], to quality assessment of linear data, considers a test data set with the geospatial information to assess and, a reference data set with the information considered as real. This algorithm has two steps. Firstly, occurs an automatic point matching, in which point in reference and test data sets are matched automatically. Secondly, it is done an interactive refinement of point matches.

2.6.2.2 Line-based Assessment

The line-based algorithm [33], to quality assessment of linear data, tries to find correspondences between line segments. Therefore, vector geometry of lines is “rasterized” in order to obtain its correspondent set of “rasterized” values. The matching with the test data is performed on “rasterized” line segments and their matching lengths and displacements are measured.

Conclusions

The Open Geospatial Consortium had an important role by encouraging the development of standards which improve the access and use of geospatial data and information. The most important of those standards were presented in this chapter.

The INSPIRE directive was a main motivation for the development of spatial data infrastructures. This directive proposed network services for sharing spatial data between the various levels of public authority in the Community. There are OGC Web services that ensure the requirements of the INSPIRE network services.

Various Quality of Service dimensions (e.g. performance, reliability, capacity, availability, security, regulatory and interoperability) need to be addressed in the invocation of geographic Web services. Moreover, data quality is also a pressing issue. For instance, an OGC working group on Data Quality was recently created, with the mission to “establish a forum for describing an interoperable framework or model for OGC Quality Assurance Web Services to enable access and sharing of high quality geospatial information, improve data analysis and ultimately influence policy decisions”.

17
Despite the efforts of the OGC work group, we do not have a framework to assess the quality of geographic Web services that considers the quality of service dimensions and the quality of the spatial data delivered by the services. The owsWatch tool monitors the performance and availability, although the data quality is not considered. Therefore, this work intends to propose and develop a framework to assess the quality of geographic Web services.
Chapter 3

Requirements for Assessing the Quality of Geographic Web Services

This chapter proposes a system framework to assess the quality of geographic Web services, as well as the quality of the data that they provide.

With the proposed framework, a human operator can assess the quality of geographic Web services through the five main functions represented on the use cases of the Figure 3.1. The considered use cases are a) administrate operations, b) monitor performance and availability, c) test scalability, d) assess data quality and e) test services. The following five sub-sections detail the requirements for each of these functionalities.

Figure 3.1: Use cases of the proposed framework.
3.1 Administrating the System

Before services can be tested or monitored they need to be registered into the system. The proposed framework defines a component for the services administration, where a human operator can register or delete services and configure their details (name, URL, type and version). Moreover, given the importance of geographic catalogue services in the discovery and publishing of geographic Web services, the human operator may want to query a catalogue service and register the services returned. Therefore, the following requirements for the proposed solution were identified:

- **RA1**: Support the registry of services (including editing of the details and deleting of the full record);
- **RA2**: Support the querying of geographic service catalogues, in order to register their published services.

3.2 Monitoring the Performance and Availability of Geographic Web Services

The monitoring of the performance and availability of geographic Web services has an essential role in assessing the quality of these services. Thus, in order to monitor the performance of Web services, one approach is to continuously measure the round-trip time between sending a request and receiving the response. The availability can be monitored by sending requests to verify if the service responds. Hence, the proposed framework defines a monitoring component where a human operator can configure monitoring instances by associating a configurable request to a registered service. Therefore, the following requirements for the proposed solution were identified:

- **RM1**: Support the monitoring of the performance and availability of geographic Web services, by periodically sending requests and measuring response times;
- **RM2**: Support the configuration of the monitoring function for each monitored service by defining the request to send, the time between the sending of requests and the time to wait for a response;
- **RM3**: Provide an interface to visualize the monitoring results through histograms and charts, summarizing results according to time of day, day of week, day of month, month, and year. With these charts, the human operator can deliberate about the monitoring results. For example, a histogram with the performance and availability values for the hours of a specific day, helps the human operator to know at which hours the service has a
better performance. Moreover, the human operator knows if the service is available during all day;

- **RM4**: Provide an interface to visualize the results achieved in a specific time period. This interface allows the human operator to request the monitoring results by indicating a specific time period.

- **RM5**: Support the comparison of the monitoring results with reference values specified by the operator. For example, the operator could want to compare the results with the performance and availability values specified in the context of the INSPIRE directive;

- **RM6**: Check the consistency of the responses returned by the geographic Web services. For example, when a "GetMap" request is sent, the system must verify if the service returns an image.

### 3.3 Testing the Scalability of Geographic Web Services

The scalability of geographic Web services to respond in situations of heavy volumes of requests, has an essential role in assessing the quality of this services. Hence, in order to test the scalability of the registered services, the proposed framework defines a load-testing component where the operator can simulate variations in the volume of sent requests. The following requirements for the proposed solution were identified:

- **RL1**: Execute load-tests by periodically sending request batches to geographic Web services;

- **RL2**: Support the configuration of load-tests by specifying the volume of requests, the time between the sending of requests and the type of requests;

- **RL3**: Provide an interface to visualize the load-test results. This interface should provide tables and charts with information about the number of requests, the time to respond, the success rate, the number of failures, and the sent requests.

### 3.4 Assessing Geospatial Data Quality

The assessment of geospatial data quality is an essential aspect in the assessment of the quality of geographic Web services. Thus, the proposed framework defines a services client interface where a human operator can define and send requests to services in order to analyze the results. Furthermore, specific algorithms are applied to assess the quality of the returned responses. From this perspective, rather than developing complex methods, this framework component
aims to prove that it is possible to assess geospatial data quality through a client interface which applies different algorithms to specific data.

The algorithms for assessing the data quality shall be loaded into the framework through a plugin. With this plugin the operator specifies the input and output parameters of the algorithm, describes the algorithm and indicates the specific resource (e.g. a Java class) that implements the algorithm. As an example of an algorithm that could be added to the framework, we could consider a process to measure the resolution with which the geographic data is presented.

To assess the geospatial data quality of geographic Web services, the following requirements were identified:

- **RD1**: Support the addition of new data quality assessment algorithms;
- **RD2**: Apply geospatial data quality algorithms to the data returned by geographic Web services;
- **RD3**: Provide an interface to visualize the results of the applied algorithms.

### 3.5 Testing Geographic Web Services

The solutions to assess the quality of geographic Web services, proposed in the preceding sections, focus on the assessment of QoS parameters and geospatial data quality criteria. The operator may also want to verify by himself the responses given by a geospatial Web service. For example, consider the following hypothetical scenario. An operator wants to decide between two Web feature services to use in his Web application. Therefore, he sends “GetFeature” requests to both services. Instead of visualizing the XML-based response that is returned by the service, the user wants to analyze the features displayed over a map layer. This hypothetical scenario shows the need to propose a testing playground for letting operators test and see the results of their interactions with geographic Web services. By analyzing this hypothetical scenario, the following requirements were identified:

- **RP1**: Support the configuration of requests to send to geographic Web services;
- **RP2**: Send requests to geographic Web services in order to assess the quality of their responses;
- **RP3**: Visualize responses returned by geographic Web services, over reference maps;
- **RP4**: Give example of requests in order to facilitate the users task;
- **RP5**: Check the logical consistency of XML responses in order to give a first and automatic assessment of the returned data.
Conclusions

This chapter lists the requirements for a framework to assess the quality of geographic Web services. The framework comprises five functions for which the requirements are described in the sub-sections of this chapter. The considered main functions are: a) register services, b) monitor performance and availability, c) test scalability, d) assess data quality, and e) test services.

The following chapter presents the implementation of the framework proposed in this chapter, describing a Web-based application called GeoWatchDog.
Chapter 4

The GeoWatchDog Application

The preceding section identified the requirements of an application for assessing the quality of geographic Web services. The application should offer functionalities to a) register services, b) monitor services, c) test service scalability, d) assess data quality and e) test services through a testing playground. In order to accomplish the identified requirements, a prototype was developed in the context of this work, namely the GeoWatchDog (GWD).

The GWD application was built over the owsWatch (see 2.5.3), a Web-based application distributed with the deegree framework to monitor the performance and availability of different OGC Web services. Furthermore, the GWD application added to the monitoring component of the owsWatch a graphing interface for visualizing the results. Besides the monitoring component, the GWD application developed one new component for each identified functionality.

Taking as basis the owsWatch architectural design, the GWD implements the Model 2 [36] architecture which, integrating the use of both servlets and JSP pages, promotes the use of the Model-View-Controller (MVC) [37] design pattern for the development of Web-based applications.

Figure 4.1 presents the components diagram of the GWD application, following the MVC design pattern. The model is constituted by a set of files that store the data of the GWD application. The GWD Portal represents the presentation layer by providing a set of interfaces that allow the interaction between a human operator and the GWD application. The controller implements the behavior of the GWD components which ensure the functionalities of the GWD application. The GWD components are the following:

- **GWD Administrator** component;
- **GWD Monitor** component;
- **GWD Load-Tester** component;
The following sub-sections detail the MVC layers that constitute the architecture of the GWD application.

4.1 GWD Model

The GWD model is based on the owsWatch model, where the information is held in XML, text and HTML files. We have the following files in the GWD Model:

- Configuration file – specifies the service types supported by the GWD application, as well as other configuration aspects such as login and files location,
- Services file – stores information about the services registered in the GWD application and their associated requests. This file also describes the monitoring instance associated to each registered service by specifying the monitoring parameters;
- Request files – XML requests for each service type supported by the GWD application and for each request type;
- Test-plan files – test-plans to be used by the GWD Load-Tester component;
• Service URL files – text files with URL’s of services which are supported by the GWD application.

The essential information to run the GWD application is held in the configuration and services files described in the following two sub-sections.

4.1.1 Configuration File

Figure 4.2 shows a sample of a configuration file that specifies the support for a WFS. The GWD support for more services can be configured adding a new “Service” element in the configuration file. The XML elements, and the respective attributes, that describe the support of a service type are as follows:

• Service – specifies the service type and version;

• ServiceRequest – within the “Service” element, defines a supported request by specifying the request name and the HTTP methods which can be used to execute the request;

• GetForm – within the “ServiceRequest” element, locates the HTML snippet form associated to the service request;

• PostForm – within the “ServiceRequest” element, locates an example request (XML request);

• RequestParameters – within the “ServiceRequest” element, defines the request parameters.

```xml
<wc:Service type="OGC:WFS" version="1.1.0">
  <wc:ServiceRequest name="GetCapabilities" isPostable="1" isGetable="1">
    <wc:GetForm>./request_snippets/all_getCapabilities_get.html</wc:GetForm>
    <wc:PostForm>../requests/wfs/Demo/GetCapabilities/xml/GetCapabilities.xml</wc:PostForm>
  </wc:ServiceRequest>
  <wc:ServiceRequest name="GetFeature" isPostable="1" isGetable="1">
    <wc:GetForm>./request_snippets/wfs_110_getFeature_get.html</wc:GetForm>
    <wc:RequestParameters>
      <wc:Key>NAMESPACE</wc:Key>
      <wc:Key>TYPENAME</wc:Key>
    </wc:RequestParameters>
  </wc:ServiceRequest>
</wc:Service>
```

Figure 4.2: Sample of the configuration file.

4.1.2 Services File

Each service registered in the GWD application has a request associated to it, which could be used by the GWD components in order to facilitate the GWD operator tasks. For example,
when the GWD operator configures the service monitoring function, the request associated to the service is automatically defined by the GWD Monitor component as the request to be sent periodically. Although, the GWD operator can define another request for the monitoring function.

Figure 4.3 shows a sample of a services file that registers a WFS with a “GetFeature” request associated. When a service is added, edited or deleted using the GWD Portal, the GWD controller automatically updates the services file. The XML elements, and its respective attributes, to describe a service within the services file are as follows:

- Service – specifies the service identifier;
- Service name – within the “SERVICE” element, indicates the service name;
- HTTP method – within the “SERVICE” element, indicates the HTTP method of the request associated to the service. In case of the HTTP Get, the inner elements are the KVP necessary to execute the request. Otherwise, in case of HTTP Post, the inner element is the XML request to post. In both cases, it is necessary to specify the service type, version and request;
- Online resource – within the “SERVICE” element, indicates the service address;
- Active – within the “SERVICE” element, indicates if the service is periodically invoked (monitored);
- Timeout – within the “SERVICE” element, indicates the time to wait for a response from the service;
- Interval – within the “SERVICE” element, indicates the time between the sending of requests when the service is active;

```xml
<watch:SERVICE id="33">
  <watch:SERVICENAME>SNIG CAOP Continente</watch:SERVICENAME>
  <watch:HTTPMETHOD type="GET">
    <watch:SERVICE>WFS</watch:SERVICE>
    <watch:VERSION>1.0.0</watch:VERSION>
    <watch:REQUEST>GetFeature</watch:REQUEST>
    <watch:NAMESPACE/>
    <watch:_TYPENAME>Distritos</watch:_TYPENAME>
  </watch:HTTPMETHOD>
  <watch:ACTIVE>true</watch:ACTIVE>
  <watch:TIMEOUT>120</watch:TIMEOUT>
  <watch:INTERVAL>1</watch:INTERVAL>
</watch:SERVICE>
```

Figure 4.3: Sample of the services file.
4.1.3 Request Files

The request files are organized into file directories organized by request type and service type. For example, Figure 4.4 shows the structure of the GWD requests directory where it is being chosen the “verticesGeodesicosPT.xml” request, which request type is “GetFeature” and it is supported by the “wfs” service type. Figure 4.5 shows the “verticesGeodesicosPT.xml” request. Requests can be added by storing them in the respective service type and request type directories.

![GWD requests directory](image)

**Figure 4.4:** GWD requests directory.

```xml
<wfs:GetFeature xmlns:wfs="http://www.opengis.net/wfs"
    xmlns:ogc="http://www.opengis.net/ogc"
    xmlns:gml="http://www.opengis.net/gml"
    xmlns:cfg="http://www.opengis.net/cite/geometry"
    outputFormat="GML2"
    version="1.0.0" service="WFS">
    <wfs:Query typeName="vg"/>
</wfs:GetFeature>
```

**Figure 4.5:** Sample of a XML request.

4.1.4 Test-Plan Files

These files define the test-plans used by the GWD Load-Tester component through a complex XML schema that specifies the configuration of the load-test plans.
4.1.5 Service URL files

Each service URL’s file provides a list of URL’s for a specific service type which is specified in
the file name. Figure 4.6 shows a o service URL’s file for the service type CSW.

http://62.48.187.117/gpt/csw202/discovery
http://idee.unizar.es/csw/servlet/cswservlet

Figure 4.6: Sample of a file with URL for the CSW type.

4.2 GWD Controller

The GWD controller implements the logic and the functionalities of the GWD application. GWD
application has the following components:

- **GWD Administrator** component responsible for services administration;
- **GWD Monitor** component responsible for services monitorization;
- **GWD Load-Tester** component responsible for testing the services scalability;
- **GWD Data-Tester** component responsible for data quality assessment;
- **GWD Playground** component responsible for testing the services.

To implement the above components, the controller implements the following functionalities:

- Process the actions of the GWD operator;
- Forward the results to the appropriate GWD views;
- Send requests to geographic services and receive the responses;
- Save the results produced by the GWD components in log files;
- Load model files.

The following sub-sections describe each one of the GWD components.

4.2.1 GWD Administrator Component

The GWD Administrator component supports the administrating operations in the GWD ap-
application by providing the following functions:
• Services administration by registering, editing and deleting services in the GWD application;

• Querying of geographic service catalogues in order to register their published services.

Figure 4.7 depicts the activities of the GWD Administrator component where the actions of the GWD operator triggers the component functions. For the services administration functions, the GWD Administrator component receives an action that defines the function requested by the GWD operator. Given the service details specified by the GWD operator, the GWD Administrator component updates the services file by deleting, editing or registering a new service in the services file.

For the registration of services published by geographic catalogues, the GWD Administrator component provides a service client interface where the GWD operator specifies a catalogue among the supported catalogue types. Moreover, the GWD operator defines a catalogue query by choosing one from a list of example queries or defining a XML request. Given the catalogue details and the query, the GWD Administrator component executes the request and returns a list with the fetched services to the operator. Finally, the GWD operator selects the services to register and defines the service details in order to register the new service.

4.2.2 GWD Monitor Component

The GWD Monitor component supports the monitoring of geographic Web services performance and availability. Based on the monitoring function of the owsWatch, the GWD Monitor component provides the following functions:

• Monitoring the performance and availability of geographic Web services;

• Configuration of services monitoring;

• Individually invocation of services;

• Visualization of the monitoring results.

Figure 4.8 depicts the activities of the GWD Monitor component which is based on the monitoring function of the owsWatch. Therefore, given the services held in the services file, a thread is responsible for verifying the refresh rates of each active service and, in the case of that time has passed, a new thread is started to send the service request. Given the response to the request, the performance and the availability results are logged in files. For the individually invocation of services, the operator can request the sending of an individual service request.

For the configuration of the services monitoring, the GWD operator specifies the monitoring details and the GWD Monitor component updates the services file.
Figure 4.7: Activity diagram of the GWD Administrator component.
Figure 4.8: Activity diagram of the GWD Monitor component.
4.2.3 GWD Load-Tester Component

The GWD Load-Tester component supports the testing of geographic Web services scalability. The GWD Load-Tester component provides the following functions:

- Specification and execution of load tests;
- Visualization of the load tests results.

Figure 4.9 depicts the activities of the GWD Load-Tester component where the GWD operator actions triggers the component actions. For load-testing geographic Web services, the GWD operator defines a test-plan by specifying the volume of requests to send, the time between the sending of requests and the requests type. Give the test-plan specified by the GWD operator, the test-plan is loaded into a “jmx” file, then the GWD Load-Test component invokes the Apache JMeter\(^1\), a load-testing tool to execute load tests. Once the load-test is executed, the GWD Load-Tester component logs the results and presents them to the operator.

For visualizing and assessing load-tests results, the GWD operator selects the service for which he wants to see the results and the GWD Load-Tester presents them.

4.2.4 GWD Data-Tester Component

The GWD Data-Tester component supports the assessing of geospatial data quality delivered by geographic Web services.

Figure 4.10 depicts the activities of the GWD Data-Tester component where the GWD operator actions triggers the component actions. As a service client interface, the Data-Tester component allows the sending of requests to the services. Given the service and the request details specified by the operator, the GWD Data-Tester component sends the request and applies an algorithm to assess the quality of the returned responses. Once the algorithm is applied, the GWD Data-Tester component logs the results and provides organized information in order to facilitate the deliberation about the achieved results.

The GWD Data-Tester component was designed to allow the adding of quality algorithms through a plugin. The following two sub-sections describes two algorithms developed in the context of this work, which could be added to the GWD Data-Tester component.

4.2.4.1 Assessing Geographic Coordinates Resolution

The GWD Data-Tester component implements the following algorithm to measure the resolution of the feature coordinates returned by a WFS in response to “GetFeature” requests:

---

\(^1\)http://jakarta.apache.org/jmeter
Figure 4.9: Activity diagram of the GWD Load-Tester component.
Figure 4.10: Activity diagram of the GWD Data-Tester component.
1. Given a response from a WFS all the elements with GML coordinates are fetched;

2. Each of the elements with GML coordinates is inspected in order to find the last non-zero
   algorism in the decimal part of the number, and the first non-zero in the integer part of
   the number;

3. The average is computed, as well as the minimum and maximum values, and the standard
   deviation for resolution (i.e., number of decimal ones).

This algorithm also defines the variable "samples", which specifies the number of GML coordi-
nates considered when the algorithm is applied.

4.2.4.2 Assessing Map Images Resolution

The GWD Data-Tester component implements an algorithm to measure the resolution or detail
of map images returned by a WMS in response to “GetMap” requests. The algorithm uses the
following notation:

- \( w \) - image width;
- \( h \) - image height;
- \( b \) - bounding box that limits the map image. The algorithm assumes that the bounding
  box is defined with coordinates in the WGS-84 system;
- \( d \) - the value for which the image will be divided;
- \( k \) - number of image samples (sub-images) considered by the algorithm for each iteration.

Given a map image with \( w \times h \) pixels and limited by \( b \), the algorithm finds the maximum level
of detail for which it is possible to visualize information on the map. The value of \( d \) and \( k \) is
defined by the GWD operator. The algorithm steps are as follows:

1. Divide the image into \( d \times 2 \) sub-images of \( \frac{w}{d} \times \frac{h}{d} \) pixels;

2. Check the value of \( k \);
   (a) Case \( k > 0 \), aleatory pick one of the sub-images obtained in step 1 and follows step
       3;
   (b) Case \( k = 0 \), it means that all the sub-images were checked and any of them has pixels
       of different colors. Follows step 4;

3. Check if the sub-image has pixels of different colors;
   (a) In positive case, hold the information about the checked sub-image and follows step
       1 with \( d = d \times 2 \);
   (b) In negative case, follows step 2 with \( k = k - 1 \);
4. Compute the geographic coordinates of the last checked sub-image with pixels of different colors;

5. Compute the ellipsoidal distance, using Vincenty’s formula [38], between the upper-left corner and the lower-right corner of the sub-image. The ellipsoidal distance defines the geographical distance between the two points on the Earth. This distance correspond to the maximum real distance for which it is possible to visualize information on the map;

### 4.2.5 GWD Playground Component

The GWD Playground component allows the testing and invoking of geographic Web services through a service client interface that provides the following functions:

- Test and invoke geographic Web services by sending requests;
- Visualize the service responses.

Figure 4.11 depicts the GWD Playground component activity. Given a service and a request specified by the GWD operator, the GWD Playground component sends the request to the service and presents the returned response. Furthermore, the services responses are presented to the operator through XML documents and when their contents allow it, the responses are presented over reference maps. For presenting services responses over maps, the GWD Playground component uses OpenLayers\(^2\), a JavaScript library for displaying map data in Web browsers, which implements WMS and WFS clients.

### 4.3 GWD View

The GWD views are a set of JSP pages which retrieve beans created by the GWD controller and presenting the information to the users. Moreover, the GWD view layer recognizes the user’s actions, such as registering a new service, and sends requests with an associated action to be executed by the GWD controller.

Although each GWD component has its specific interface, there are three common views which are used by all of them. These are the services list, the service details, and the request details.

Figure 4.12 presents the services list view which shows the details of the services registered in the GWD application. The services list presents service details such as:

- Name — the service name;
- Type — the service type;

\(^2\)http://openlayers.org
Figure 4.11: Activity diagram of the GWD Playground component.
• Version — the service type version;
• Request — the request associated to the service;
• Select — a HTML radiobox to select the service in order to activate the actions over it.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Version</th>
<th>Request</th>
<th>Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>WMS demo Cape</td>
<td>WMS</td>
<td>1.1.1</td>
<td>GetCapabilities</td>
<td>○</td>
</tr>
<tr>
<td>SNIG CAOF Continente 2006.1</td>
<td>WMS</td>
<td>1.1.1</td>
<td>GetCapabilities</td>
<td>○</td>
</tr>
<tr>
<td>IDEE Cartocidad: Nomenclatura de Código postal</td>
<td>WFS</td>
<td>1.1.0</td>
<td>GetCapabilities</td>
<td>○</td>
</tr>
<tr>
<td>WMS demo GetMap</td>
<td>WMS</td>
<td>1.1.1</td>
<td>GetMap</td>
<td>○</td>
</tr>
<tr>
<td>CSWGetRecords test</td>
<td>CSW</td>
<td>2.0.2</td>
<td>GetRecords</td>
<td>○</td>
</tr>
<tr>
<td>WCS demo GetCoverage</td>
<td>WCS</td>
<td>1.0.0</td>
<td>GetCoverage</td>
<td>○</td>
</tr>
<tr>
<td>SNIG - Verdes Geodesicos Portugal Continental</td>
<td>WMS</td>
<td>1.1.1</td>
<td>GetMap</td>
<td>○</td>
</tr>
</tbody>
</table>

**Figure 4.12:** Services list view.

Since a service registered in the GWD application has an associated request, the service details and the request details views are always used in conjunction. These views show the respective details and allow the GWD operator to edit them. The GWD components use these views for different tasks, such as, edit a registered service or send a request to the service.

Figure 4.13 presents the service details view where is being presented the details of a WMS 1.1.0, called ‘ESRI Map”. This view shows the service details, namely, the name, the type, the version and the URL.

**Figure 4.13:** Service details view.

Figure 4.14 presents the request details view where is being presented the details of a “GetMap” request. The request details are as follows:
• Request Type – the request type which depends on the service type and its version;
• HTTP Method – the HTTP Method to use in case of sending the request (Get or Post);
• Request Details – the request parameters which depends on the HTTP method. In the case of a Get request, it specifies the request parameters which are defined in the request snippet associated to the request type. In the case of a Post request, it specifies the XML request.

![Request Details Table]

Figure 4.14: Request details.

The HTML snippet forms, used by the GWD views for presenting the details of a specific service, are stored in a directory organized by request type. Figure 4.15 shows a sample of a snippet form for the “GetFeature” request, which specifies two “input” fields for the “GetFeature” parameters ("Typename" and "Namespace").

The following sub-sections present the interfaces specific to each GWD component.

### 4.3.1 GWD Administrator Interface

The interface of the GWD Administrator component allows the GWD operator to execute the administration functions. Figure 4.16 presents the interface of the GWD Administrator component which presents the list of the registered services. For each entry in the registered services list, the operator can execute the following actions:

• Edit the service details by selecting the service and clicking on the button “Edit”;
Figure 4.15: HTML snippet form of the WFS “GetFeature”.

- Delete a registered service by selecting the service and clicking on the button “Delete”.

The operator can also execute the following actions:

- Register a new service by clicking on the button “Register New”;
- Send queries to geographic catalogues in order to register their published services. For that purpose the GWD operator clicks in the button “Catalogue Client”;

Figure 4.16: Interface of the GWD Administrator component.

Figure 4.17 shows a scenario where a WMS is being registered with an associated “GetMap” request.
Figure 4.17: Interface of the GWD Administrator component to register a service.
Figure 4.18 presents the catalogue client interface of the GWD Administrator where a query is being defined for getting, from a CSW catalogue, all the services that have in their descriptions the text “WMS”. Figure 4.19 shows the published services returned by this query. The returned services can be registered in the GWD application by selecting them and clicking in the button “Register”.

![Add Catalogue Services](image)

**Figure 4.18**: Catalogue client interface.

### 4.3.2 GWD Monitor Interface

The GWD Monitor interface allows the operator to execute the GWD Monitor functions. Figure 4.20 presents the GWD Monitor interface which presents the monitoring list filled with the services that can be monitored. For each entry in the monitoring list, the user can execute the following actions:

- Configure the service monitoring by clicking in the button “Edit Monitor”;
- Invoke the service by clicking on the button “Invoke” to send an individual request;
- See the monitoring results of the service by clicking on the button “Performance Results”;
- See the service monitoring details by selecting the service.

Furthermore, each entry in the monitoring list has the following attributes:

- Name – the service name;
- Type – the service type;
Figure 4.19: Services returned by a catalogue.

- Request – the request type which is sent by the monitoring function;
- Monitor – an icon indicating if the service is being monitored. A green circle for the positive case, otherwise a red circle;
- Performance – the performance of the last test executed;
- Availability – the service availability relative to the last test executed;
- Last Test – the time of last test executed;
- Select – a HTML “radiobox” to select the service in order to activate the actions over it and see the service details.

Figure 4.21 presents the GWD Monitor interface for configuring the services monitoring, where a “GetMap” request is being defined to be periodically sent to a WMS through a HTTP Get method, in periods of one minute. The monitored services configuration is done by setting the request details and the following parameters related to the monitoring function:

- Monitor – activate the service monitoring;
- Refresh Rate – specify the period, in minutes, between the sending of two requests;
- Server Timeout – indicate, in seconds, the server timeout.
The monitoring results presentation is a specific improvement to the owsWatch monitoring function. Namely, instead a simple results table, the GWD Monitor component provides a graphic interface to visualize the monitoring results.

Figure 4.22 presents the monitoring results for the service “WCS demo GetCoverage”, where is being showed a table with the global results indicating the number of requests sent to the service, the average performance and the availability percentage. Furthermore, it is being presented a graph with the average performance per hour of the day 12-09-2009.

The graphic interface was developed with the Bluff3, a JavaScript graphing library. The graphic interface allows the following actions:

- Visualize the performance results through a graph organized by different time series. Namely, it is possible to see the average performance per hours of day, per days of week, per days of month, per month and per year;

- Visualize the performance results related to a specific time period specified by the user;

- Compare the measured values with reference values specified by the user.

Figure 4.20: Monitoring list.

---

**Monitoring List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Request</th>
<th>Monitor</th>
<th>Performance</th>
<th>Availability</th>
<th>Last Test</th>
<th>Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>WMS demo Capa</td>
<td>WMS</td>
<td>GetCapabilities</td>
<td></td>
<td>avg</td>
<td></td>
<td>Not yet tested</td>
<td></td>
</tr>
<tr>
<td>WFS demo Capa</td>
<td>WFS</td>
<td>GetCapabilities</td>
<td></td>
<td>avg</td>
<td></td>
<td>Not yet tested</td>
<td></td>
</tr>
<tr>
<td>WCS demo Capa</td>
<td>WCS</td>
<td>GetCapabilities</td>
<td></td>
<td>avg</td>
<td></td>
<td>Not yet tested</td>
<td></td>
</tr>
<tr>
<td>CSW demo Capa</td>
<td>CSW</td>
<td>GetCapabilities</td>
<td></td>
<td>avg</td>
<td></td>
<td>Not yet tested</td>
<td></td>
</tr>
<tr>
<td>WMS demo GetMap</td>
<td>WMS</td>
<td>GetMap</td>
<td></td>
<td>avg</td>
<td></td>
<td>Not yet tested</td>
<td></td>
</tr>
<tr>
<td>WFS demo GetFeature</td>
<td>WFS</td>
<td>GetFeature</td>
<td></td>
<td>avg</td>
<td></td>
<td>Not yet tested</td>
<td></td>
</tr>
<tr>
<td>CSWGetRecords test</td>
<td>CSW</td>
<td>GetRecords</td>
<td></td>
<td>avg</td>
<td></td>
<td>Not yet tested</td>
<td></td>
</tr>
</tbody>
</table>

3http://bluff.jcoglan.com
Figure 4.21: Interface to configure the monitoring function.
Figure 4.22: GWD Monitor results.
4.3.3 GWD Load-Tester Interface

The GWD Load-Tester interface allows the operator to execute the GWD Load-Tester functions. For each entry in the services list of the GWD application, the GWD operator can execute the following actions:

- Specify a test plan by selecting a service and clicking the button “Test Service”;
- See the historic of executed load tests by selecting a service and clicking the button “Results”;

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Scalability</th>
<th>Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>WMS demo Capa</td>
<td>WMS</td>
<td>value</td>
<td>⬜</td>
</tr>
<tr>
<td>SNIG CAOP Continental 2008.1</td>
<td>WMS</td>
<td>value</td>
<td>⬜</td>
</tr>
<tr>
<td>IDEE Certificado:</td>
<td>WFS</td>
<td>value</td>
<td>⬜</td>
</tr>
<tr>
<td>Nomenclatura de Codigo postal</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WMS demo GetMap</td>
<td>WMS</td>
<td>value</td>
<td>⬜</td>
</tr>
<tr>
<td>WFS demo GetFeature</td>
<td>WFS</td>
<td>value</td>
<td>⬜</td>
</tr>
<tr>
<td>CSWGetRecords test</td>
<td>CSW</td>
<td>value</td>
<td>⬜</td>
</tr>
<tr>
<td>WCS demo GetCoverage</td>
<td>WCS</td>
<td>value</td>
<td>⬜</td>
</tr>
<tr>
<td>SNIG - Vertices Geodeticos</td>
<td>WMS</td>
<td>value</td>
<td>⬜</td>
</tr>
<tr>
<td>Portugal Continental</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.23: GWD Load-Tester services list.

Figure 4.24 shows a load test-plan which simulates ten users to send “GetCapabilities” requests to a WFS. The requests are sent in three iterations in time intervals of three seconds. As a load-tests interface, the GWD Load-Tester allows the specification of test-plans by defining the request details and the following parameters related to the simulation of a heavy load on the service:

- Test name;
- Number of threads (simulated users);
- Loop count (number of iterations) ;
- Period between each loop.

Figure 4.25 shows the GWD Load-Tester interface for presenting test-plan results.
Figure 4.24: GWD Load-Tester test-plan.
4.3.4 GWD Data-Tester Interface

The GWD Data-Tester interface allows the GWD operator to execute the GWD Data-Tester functions.

Figure 4.26 presents the algorithms list where the operator can select one to assess the geospatial data. Figure 4.27 presents the service client interface where a “GetFeature” request to be sent to a WFS is being defined. Figure 4.28 presents the results achieved by applying the quality assessment algorithm to measure the coordinates resolution.

Figure 4.25: GWD Load-Tester results.

Figure 4.26: Algorithms list.
Figure 4.27: Service client interface of the GWD Data-Tester component.

Figure 4.28: Results of the GWD Data-Tester component.
4.3.5 GWD Playground Interface

The interface of the GWD Playground component is a service client interface which allows the GWD operator to execute the GWD Playground functions. Figure 4.29 presents the interface of the GWD Playground component where a map, returned by a WMS, is being presented over a reference map. In this interface the GWD can freely test geographic Web service by editing the requests and seeing their responses. To facilitate the GWD operator task, the interface provides a example request list with editable requests.

Conclusions

This chapter presented the GeoWatchDog (GWD) application, which implements the requirements proposed in the previous chapter. The monitoring function of the GWD Monitor component is based on the owsWatch. Although the GWD Monitor added a new graphing interface for visualizing the monitoring results. Besides the monitoring component, the GWD application developed one new component for each identified functionality.

The sections of this chapter describe the architecture of the GeoWatchDog application which is based on the owsWatch architecture. The GWD architecture implements the MVC design pattern for the developing of Web-based applications. The model is a set of files, which hold the essential information to run the GWD application. The controller implements the GWD behaviour. The views is a set of interfaces which allow the interaction between the human operator and the GWD application.

In order to assess the GWD behavior in a real context, the following chapter presents several experiments to evaluate the GWD ability to assess the quality of geographic Web services.
Figure 4.29: Interface of the GWD Playground component.
Chapter 5

Evaluation

This chapter describes the experiments made in order to evaluate the ability of the GWD application for assessing the quality of geographic Web services. The experiments used services available from a) the Portuguese Geographic Institute (IGP), b) the Spanish IDEE portal and c) others available on the Web. Given these services, the behaviour of the GWD components were evaluated by following the dynamic analysis method [39].

Before conducting the experiments described in the following sections, a set of services was registered in the system using the GWD Administrator component.

Each registered service has an associated request that is used by the GWD components to realize their functionalities. Table 5.1 lists the services manually registered in the system. For each service in this table, we have its name, type, version and the type of the associated request.

<table>
<thead>
<tr>
<th>Service Name</th>
<th>Type</th>
<th>Version</th>
<th>Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metacarta</td>
<td>WMS</td>
<td>1.1.0</td>
<td>GetMap</td>
</tr>
<tr>
<td>SNIRH Aguas Subterraneas</td>
<td>WMS</td>
<td>1.1.1</td>
<td>GetMap</td>
</tr>
<tr>
<td>SNIRH Aguas Superficiais</td>
<td>WMS</td>
<td>1.1.1</td>
<td>GetMap</td>
</tr>
<tr>
<td>MassGIS</td>
<td>WFS</td>
<td>1.1.0</td>
<td>GetFeature</td>
</tr>
<tr>
<td>Sigma</td>
<td>WFS</td>
<td>1.0.0</td>
<td>GetFeature</td>
</tr>
<tr>
<td>Deegree</td>
<td>WCS</td>
<td>1.0.0</td>
<td>GetCoverage</td>
</tr>
<tr>
<td>Digmap Gazetteer</td>
<td>ADLGP</td>
<td>1.2</td>
<td>GazetteerQuery</td>
</tr>
</tbody>
</table>

Table 5.1: Services registered in the GWD application.

The associated requests depend on the service type, therefore Table 5.2 presents the “GetMap” requests associated to the WMS implementations listed on Table 5.1. Each request in this table asks for a map image or images (layers), by specifying the size (width and height), the format and the transparency (transparent) of the image. The requested images correspond to the earth surface determined by a set of geographic coordinates (BBOX), which are specified in a spatial referencing system (SRS) defined as a EPSG code.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Metacarta</th>
<th>SNIRH Aguas Subterraneas</th>
<th>SNIRH Aguas Superficiais</th>
</tr>
</thead>
<tbody>
<tr>
<td>Width</td>
<td>400</td>
<td>800</td>
<td>400</td>
</tr>
<tr>
<td>Height</td>
<td>300</td>
<td>600</td>
<td>300</td>
</tr>
<tr>
<td>Layers</td>
<td>basic</td>
<td>15,14,13</td>
<td>13</td>
</tr>
<tr>
<td>Transparent</td>
<td>False</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>Format</td>
<td>image/png</td>
<td>image/png</td>
<td>image/jpeg</td>
</tr>
<tr>
<td>SRS</td>
<td>EPSG:4326</td>
<td>EPSG:4326</td>
<td>EPSG:4326</td>
</tr>
<tr>
<td>BBOX</td>
<td>-180.0, -90.0, 180.0, 90.0</td>
<td>-10.7800952637, 36.6757114678, -5.9433072076, 42.3913128613</td>
<td>-9.6128206501, 36.9466769059, -6.1720078879, 42.1410521432</td>
</tr>
</tbody>
</table>

Table 5.2: Parameters of the “GetMap” requests.

In the case of the “MassGIS”, a WFS implementation, its associated request (“GetFeature”) asks for all the geographic features of the type “massgis:DMARRIER.RESBUFF1000”.

The “Sigma” was the other WFS registered in the GWD application. This service also has a “GetFeature” request associated, although, the request is XML-encoded (to be used with the HTTP post method). Figure 5.1 presents the request associated to the “Sigma”.

```xml
<wfs :GetFeature
  xmlns : wfs="http ://www. opengis .net / wfs"
  xmlns : ogc="http ://www. opengis .net / ogc"
  xmlns : gml="http ://www. opengis .net /gml"
  xmlns : c g f="http ://www. opengis .net / cite / geometry"
  outputFormat="GML2" version="1.1.0" service="WFS">
  <wfs : Query typeName="states">
  </wfs : Query>
</wfs :GetFeature>
```

Figure 5.1: “GetFeature” request for the Sigma.

In the case of the WCS provided by the Deegree framework, its associated request asks for an image representing a coverage called “saltlakecity”. Table 5.3 lists the parameters of the “GetCoverage” request. This request asks for a coverage image which is identified with the same parameters of the WMS specification for map images, excepting the transparency parameter.
Table 5.3: Parameters of the “GetCoverage” request for the Deegree WCS.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Width</td>
<td>500</td>
</tr>
<tr>
<td>Height</td>
<td>500</td>
</tr>
<tr>
<td>Layers</td>
<td>saltlakecity</td>
</tr>
<tr>
<td>Format</td>
<td>jpeg</td>
</tr>
<tr>
<td>CRS</td>
<td>EPSG:26912</td>
</tr>
<tr>
<td>BBOX</td>
<td>420857.5, 4504382.5, 431410.5, 4518364.5</td>
</tr>
</tbody>
</table>

The “Digmap Gazetteer” was registered and associated to a “GazetterQuery” request. Figure 5.2 presents the XML document that defines this request.

```xml
<gazetteer-service
    xmlns="http://www.alexandria.ucsb.edu/gazetteer"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.alexandria.ucsb.edu/gazetteer
    http://www.alexandria.ucsb.edu/gazetteer/protocol/gazetteerservice.xsd"
    version="1.2">
    <query-request>
        <gazetteer-query>
            <and>
                <name-query operator="contains-phrase" text="las vegas"/>
                <class-query thesaurus="ADL Feature Type Thesaurus"
                    term="populated places"/>
            </and>
        </gazetteer-query>
    </query-request>
</gazetteer-service>
```

Figure 5.2: “GazetterQuery” request for the “Digmap Gazetteer”.

5.1 Registration of Services published by a Catalogue

The objective of this experiment was to evaluate the GWD ability in terms of querying a catalogue service and registering their published services. The experiment consisted in the following steps:

1. Query the geographic services catalogue provided by the Geographic Information National System in Portugal\(^8\) (SNIG), in order to get all the published WMS and WFS;

2. Register the services returned by the catalogue and associate a request;

\(^8\)http://62.48.187.117/gpt/csw202/discovery
The SNIG catalogue implements the CSW specification. Since the CSW does not support queries by service type, the sent queries check if the catalogue records has the text correspondent to the requested service type (“WMS” and “WFS”) in their descriptions. Figure 5.3 presents the sent query sent in order to fetch all the records that contain a specific text. In this case, the text is “WMS”.

```xml
<csw:GetRecords service="CSW" version="2.0.2"
outputFormat="application/xml"
outputSchema="http://www.opengis.net/cat/csw/2.0.2"
resultType="RESULTS"
maxRecords="20"
xmims:csw="http://www.opengis.net/cat/csw/2.0.2"
xmims:ogc="http://www.opengis.net/ogc"
xmims:apiso="http://www.opengis.net/cat/csw/apiso/1.0">
  <csw:Query typeNames="csw:Record">
    <csw:ElementSetName>full</csw:ElementSetName>
    <csw:Constraint version="1.1.0">
      <ogc:Filter>
        <ogc:PropertyIsLike wildCard="\%" singleChar="\_" escape="/">
          <ogc:PropertyName>anyText</ogc:PropertyName>
          <ogc:Literal>%wms%</ogc:Literal>
        </ogc:PropertyIsLike>
      </ogc:Filter>
    </csw:Constraint>
  </csw:Query>
</csw:GetRecords>
```

**Figure 5.3:** Query sent to the SNIG catalogue.

Ideally, the services returned by the catalogue should match the services listed in the Table 5.4, which lists the WMS and WFS published by the SNIG catalogue. Besides the service details, this table also shows if a specific service was returned by the catalogue service.

<table>
<thead>
<tr>
<th>Service</th>
<th>Type</th>
<th>Returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAOP Continente 6.0 2008</td>
<td>WMS</td>
<td>✓</td>
</tr>
<tr>
<td>CAOP Madeira 6.0</td>
<td>WMS</td>
<td>✓</td>
</tr>
<tr>
<td>CAOP Acores 6.0</td>
<td>WMS</td>
<td>✓</td>
</tr>
<tr>
<td>CRIF 2009</td>
<td>WMS</td>
<td>✓</td>
</tr>
<tr>
<td>Carta de Portugal Continental</td>
<td>WMS</td>
<td>✓</td>
</tr>
<tr>
<td>Modelo Digital de Terreno</td>
<td>WMS</td>
<td>✓</td>
</tr>
<tr>
<td>Modelo Digital de Terreno</td>
<td>WMS</td>
<td>✓</td>
</tr>
<tr>
<td>Rede Geodesica Nacional Cotinente</td>
<td>WMS</td>
<td>✓</td>
</tr>
<tr>
<td>CAOP Continente 6.0</td>
<td>WFS</td>
<td>✓</td>
</tr>
<tr>
<td>CAOP Madeira 6.0</td>
<td>WFS</td>
<td>✓</td>
</tr>
<tr>
<td>CAOP Acores 6.0</td>
<td>WFS</td>
<td>✓</td>
</tr>
<tr>
<td>Rede Geodesica Nacional Cotinente</td>
<td>WFS</td>
<td>✓</td>
</tr>
<tr>
<td>Carta de Portugal Continental</td>
<td>WFS</td>
<td>✓</td>
</tr>
<tr>
<td>Carta de Portugal Continental</td>
<td>WFS</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Table 5.4:** WMS and WFS published by the SNIG catalogue.

As it can be seen in the Table 5.4, just one of the services published by the catalogue was not returned. A possible reason for that fact, is the non-existence of the text “WFS” in the record.
Furthermore, all the services were registered in the GWD application.

5.2 Evaluation of the Services Monitoring

The objective of this experiment was to evaluate the GWD ability to monitor the registered services. The experiment consisted in using the GWD Monitor component to activate and configure the monitoring function for the services listed in the Table 5.1. The requests periodically sent by the monitoring function were those associated to the services. For each service, the GWD operator configured the monitoring function as follows:

1. Activated the monitoring function;
2. The request periodically sent was the request associated to the service;
3. The refresh rate, which defines the time between each sent request, was one minute;
4. The timeout, which defines the time that the GWD application waits for a response from the service, was thirty seconds;

The expected result for this experiment was the monitoring of all the services in order to produce results which allows the GWD operator to assess the performance and availability of the services. Furthermore, it was expectable that the GWD operator could visualize the monitoring results through histograms, where he can consult the results for specific periods or compare them with reference values.

After two weeks monitoring the services, the results were analysed and we can concluded that all the services were monitored. Moreover, the GWD operator consulted the monitoring results, through the graphing interface provided by the GWD Monitor component. Figure 5.4 shows a graph provided by this interface where are presented the average performance values per hour of the service “SNIG CAOP Continente ” for the 31th July 2009. Analysing this graph, it is perceptible that the service performance was almost constant during the day, except between the fifteen and seventeen hours.

The interface of the GWD Monitor also permitted the user to compare the obtained results with reference values. Figure 5.5 shows a graph where are presented the average performance values of the service “Digmap Gazetteer”, for the week days. The performance values in this graph are compared with a reference value of 3000 miliseconds and it is possible to conclude that only in sundays the service had an higher performance.

Table 5.5 summarizes the monitoring results where it can be seen, for each service, the sent request, the average performance, the availability percentage and the number of sent requests. Analysing this table we can conclude the following:

• The availability values are close to 100%;
• There was a discrepancy in the performance values of the services. A possible reason to 
  this discrepancy, is the difference of bytes returned by the services, which depends on the 
  request type. For example, the “SNIRH Aguas Subterraneas”, which returns map images, 
  had higher performance values than other services, such as the “Digmap Gazetteer” which 
  returns a XML document;

• Taking into account the QoS requirements proposed by the INSPIRE directive for the 
  geographic Web services, we can conclude that all the services, except the “Deegree WCS”, 
  satisfy the requirements in terms of availability, which must be 99%;

5.3 Evaluation of the Services Load-Testing

The objective of this experiment was to evaluate the GWG ability to load test geographic Web 
  services. The experiment consisted in using the GWG Load-Tester component to specify and 
  execute load test-plans for a set of services provided by the SNIG and IDEE.

The set of tested services was constituted by two WMS (version 1.1.1), two WCS (version 2.0.2) 
  and two WFS (version 1.1.0). For each type of service, we had one service provided by the SNIG
and one provided by the IDEE.

The expected result for this experiment was the services load-testing and the producing of results, which allow the GWD operator to assess and deliberate about the services scalability. It is also expectable that the GWD operator consults the results of the load-tests in order to compare the results between the services provided by the IDEE and the SNIG.

For each service, the GWD operator configured three test-plans with the following parameters:

- Service – details of the tested service;
- Request – details of the request sent by each simulated user;
- The number of simulated users per iteration. This value was thirty for the first test, sixty for the second test, and ninety for the third test;
- The period, which defines the time between the each iteration, was two seconds;
- The number of iterations was three.

Each test-plan defines a request to be sent by the GWD application. In the case of the WMS, Ta-

**Figure 5.5:** Performance graph of the service “Digmap Gazetteer.”
<table>
<thead>
<tr>
<th>Service</th>
<th>Type</th>
<th>Request</th>
<th>Performance</th>
<th>Availability</th>
<th>Requests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metacarta WMS</td>
<td>WMS</td>
<td>GetMap</td>
<td>572</td>
<td>99%</td>
<td>21600</td>
</tr>
<tr>
<td>MassGIS WFS</td>
<td>WFS</td>
<td>GetFeature</td>
<td>687</td>
<td>99%</td>
<td>21600</td>
</tr>
<tr>
<td>Deegree WCS</td>
<td>WCS</td>
<td>GetCoverage</td>
<td>529</td>
<td>97%</td>
<td>21600</td>
</tr>
<tr>
<td>SNIRH Agua Subterraneas WMS</td>
<td>WMS</td>
<td>GetMap</td>
<td>2471</td>
<td>99%</td>
<td>21600</td>
</tr>
<tr>
<td>SNIRH Agua Superficiais WMS</td>
<td>WMS</td>
<td>GetMap</td>
<td>3007</td>
<td>99%</td>
<td>21600</td>
</tr>
<tr>
<td>Digmap Gazetteer</td>
<td>ADL-GP</td>
<td>GazetteerQuery</td>
<td>146</td>
<td>100%</td>
<td>21600</td>
</tr>
</tbody>
</table>

Table 5.5: Performance and availability values of the monitored services.

Table 5.6 specifies the parameters of the “GetMap” requests sent to the SNIG “CAOP Continente”\(^9\) and to the IDEE “Mapa Base”\(^10\).

<table>
<thead>
<tr>
<th>Parameters</th>
<th>CAOP Continente</th>
<th>Mapa Base</th>
</tr>
</thead>
<tbody>
<tr>
<td>Width</td>
<td>400</td>
<td>400</td>
</tr>
<tr>
<td>Height</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>Layers</td>
<td>CAOP</td>
<td>Relieve</td>
</tr>
<tr>
<td>Transparent</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>Format</td>
<td>image/png</td>
<td>image/png</td>
</tr>
<tr>
<td>SRS</td>
<td>EPSG:4326</td>
<td>EPSG:4326</td>
</tr>
<tr>
<td>BBOX</td>
<td>-10.1905, 36.8994, -5.71298, 42.1896</td>
<td>-3.4650655, 3.62939155, 44.13107</td>
</tr>
</tbody>
</table>

Table 5.6: Parameters of the “GetMap” requests sent in the load-test.

In the case of the CSW, both the catalogue provided by the SNIG\(^11\) and the IDEE\(^12\) were tested through “GetCapabilities” requests, for which is not necessary any parameter.

In the case of the WFS, the SNIG “Vertices Geodesicos”\(^13\) was tested through a “GetFeature” request with the “typename” parameter equal to “vg”. The IDDE “Nomenclatura Municipio”\(^14\) was tested through a “GetFeature” request with the “typename” parameter equal to “app:Entidad”.

Table 5.7 summarizes the results obtained with the services load-testing. Analysing this table, we can conclude the following:

\(^10\)http://www.idee.es/wms/IDEE-Base/IDEE-Base
\(^11\)http://62.48.187.117/gpt/csw202/discovery
\(^12\)http://idee.unizar.es/csw/servlet/cswservlet
\(^13\)http://mapas.igeo.pt/ows/vgs
\(^14\)http://www.cartociudad.es/wfs-municipio/services
• All of the sent requests were successfully executed;

• Although the size of the map images is identical, the image returned by the “Mapa Base” is bigger than that returned by the “CAOP Continente”. Probably, this is the reason for the “CAOP Continente” be faster to respond in situations of high volume of requests;

• The SNIG catalogue was faster than the IDEE catalogue to respond to the “GetCapabilities” request. A possible reason for that difference is the size of the capabilities documents which is bigger for the IDEE catalogue;

• For the services “IDEE Mapa Base”, “IDEE Catalogue”, “SNIG Vertices Geodesicos” and “SNIG CAOP Continente”, the average time to respond to the requests increased alongside of the number of tests (simulated users);

<table>
<thead>
<tr>
<th>Service</th>
<th>Tests</th>
<th>Failures</th>
<th>Average Time</th>
<th>Average Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mapa Base</td>
<td>90</td>
<td>0</td>
<td>1239</td>
<td>147756</td>
</tr>
<tr>
<td>Mapa Base</td>
<td>180</td>
<td>0</td>
<td>4632</td>
<td>146650</td>
</tr>
<tr>
<td>Mapa Base</td>
<td>270</td>
<td>0</td>
<td>4945</td>
<td>146834</td>
</tr>
<tr>
<td>CAOP Continente</td>
<td>90</td>
<td>0</td>
<td>800</td>
<td>13679</td>
</tr>
<tr>
<td>CAOP Continente</td>
<td>180</td>
<td>0</td>
<td>2246</td>
<td>13679</td>
</tr>
<tr>
<td>CAOP Continente</td>
<td>270</td>
<td>0</td>
<td>3555</td>
<td>13679</td>
</tr>
<tr>
<td>IDEE Catalogue</td>
<td>90</td>
<td>0</td>
<td>131</td>
<td>10302</td>
</tr>
<tr>
<td>IDEE Catalogue</td>
<td>180</td>
<td>0</td>
<td>484</td>
<td>15453</td>
</tr>
<tr>
<td>IDEE Catalogue</td>
<td>270</td>
<td>0</td>
<td>1331</td>
<td>18038</td>
</tr>
<tr>
<td>SNIG Catalogue</td>
<td>90</td>
<td>0</td>
<td>62</td>
<td>6033</td>
</tr>
<tr>
<td>SNIG Catalogue</td>
<td>180</td>
<td>0</td>
<td>70</td>
<td>6033</td>
</tr>
<tr>
<td>SNIG Catalogue</td>
<td>270</td>
<td>0</td>
<td>39</td>
<td>6033</td>
</tr>
<tr>
<td>Nomenclatura Municipio</td>
<td>90</td>
<td>0</td>
<td>26</td>
<td>211</td>
</tr>
<tr>
<td>Nomenclatura Municipio</td>
<td>180</td>
<td>0</td>
<td>24</td>
<td>211</td>
</tr>
<tr>
<td>Nomenclatura Municipio</td>
<td>270</td>
<td>0</td>
<td>21</td>
<td>211</td>
</tr>
<tr>
<td>Vertices Geodesicos</td>
<td>90</td>
<td>0</td>
<td>580</td>
<td>551</td>
</tr>
<tr>
<td>Vertices Geodesicos</td>
<td>180</td>
<td>0</td>
<td>3597</td>
<td>551</td>
</tr>
<tr>
<td>Vertices Geodesicos</td>
<td>270</td>
<td>0</td>
<td>5074</td>
<td>551</td>
</tr>
</tbody>
</table>

Table 5.7: Load-testing results.

5.4 Evaluation of the Data Quality Assessment

The objective of the two experiments described in the following sub-sections, was to evaluate the GWD ability to assess the data quality delivered by geographic Web services. For that purpose, the data quality algorithms provided by the GWD Data-Tester component were experimented. Ideally, both experiments should correctly apply the algorithms to the data delivered by the geographic Web services and produce results to assess and deliberate about that data.
5.4.1 Coordinates Precision

The objective of this experiment was to apply the algorithm, provided by the GWD Data-Tester component, to assess the precision of the coordinates which describes the features delivered by WFS.

The experiment consisted in using the GWD Data-Tester component to send “GetFeature” requests to the services provided by the SNIG and the IDEE. Hence, for the SNIG service “Vertices Geodesicos” the “typename” of the requested features was “vg”. For the IDEE services, the “Nomenclatura Municipio” and the “Nomenclatura Codigo Postal”, the “typename” of the requested features was “app:Entidad” and the “namespace” was “xmlns(app= http://www.deegree.org/app)”. Furthermore, the number of samples which defines the number of coordinates considered by the algorithm was an hundred.

Table 5.8 presents the average (Avg), the maximum (Max), the minimum (Min) and the standard deviation (SD) for the number of non-zero algorisms found in the integer and decimal parts of the coordinates.

<table>
<thead>
<tr>
<th>Service</th>
<th>Integer Avg</th>
<th>Integer Max</th>
<th>Integer Min</th>
<th>Integer SD</th>
<th>Decimal Avg</th>
<th>Decimal Max</th>
<th>Decimal Min</th>
<th>Decimal SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vertices Geodesicos</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td>0.0</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>0.74</td>
</tr>
<tr>
<td>Nomenclatura Municipio</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0.69</td>
<td>13</td>
<td>14</td>
<td>13</td>
<td>0.73</td>
</tr>
<tr>
<td>Nomenclatura Codigo Postal</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0.82</td>
<td>13</td>
<td>14</td>
<td>11</td>
<td>0.73</td>
</tr>
</tbody>
</table>

Table 5.8: Results of the coordinates precision algorithm.

Analysing the results of the Table 5.8, we can conclude that the WFS services provided by the IDEE return features with more resolution than those provided by the SNIG. Namely, the average of non-zero algorisms in the decimal part of the coordinates is bigger for the IDEE services.

The 5.4.2 Map Image Resolution

The objective of this experiment was to apply the algorithm, provided by the GWD Data-Tester component, to assess the resolution of map images delivered by WMS.

The experiment consisted in using the GWD Data-Tester component to send “GetMap” requests to the WMS listed in the Table 5.9. For each service in this table, the GWD operator sent three requests with different samples (ten, twenty and thirty) and saved the best resolution value.

Since all the services listed in the Table 5.9, except the ‘CAOP Acores”, were already used in the previous experiments, the requests sent were the same. Therefore, Table 5.2 describes the
requests sent to the “Metacarta”, “Aguas Subterraneas” and “Aguas Superficiais”. Table 5.6 describes the requests sent to the CAOP Continente and Mapa Base. Table 5.10 describes the request sent to the “CAOP Acores”.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>CAOP Acores</th>
</tr>
</thead>
<tbody>
<tr>
<td>Width</td>
<td>400</td>
</tr>
<tr>
<td>Height</td>
<td>300</td>
</tr>
<tr>
<td>Layers</td>
<td>CAOP2008_Acores</td>
</tr>
<tr>
<td>Transparent</td>
<td>True</td>
</tr>
<tr>
<td>Format</td>
<td>image/png</td>
</tr>
<tr>
<td>SRS</td>
<td>EPSG:4326</td>
</tr>
<tr>
<td>BBOX</td>
<td>-32.0763, 36.049, -23.6555, 40.6502</td>
</tr>
</tbody>
</table>

Table 5.10: Parameters of the “GetMap” request for the service “CAOP Acores”.

After sending the requests to the WMS, we concluded that the images returned by the services were assessed by the algorithm. Table 5.11 presents the results for each tested service, where the pixels field is the size of the minimum area of the image with pixels of different colors, and the resolution indicates, in kilometers, the real distance for which it is possible to distinguish information on the map.

<table>
<thead>
<tr>
<th>Service</th>
<th>Samples</th>
<th>Pixels</th>
<th>Resolution (km)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAOP Continente</td>
<td>20</td>
<td>$3 \times 2$</td>
<td>4.77</td>
</tr>
<tr>
<td>CAOP Acores</td>
<td>10</td>
<td>$100 \times 75$</td>
<td>212.82</td>
</tr>
<tr>
<td>Metacarta</td>
<td>10</td>
<td>$3 \times 2$</td>
<td>177.60</td>
</tr>
<tr>
<td>Aguas Subterraneas</td>
<td>20</td>
<td>$3 \times 2$</td>
<td>5.17</td>
</tr>
<tr>
<td>Aguas Superficiais</td>
<td>30</td>
<td>$3 \times 2$</td>
<td>4.38</td>
</tr>
<tr>
<td>Mapa Base</td>
<td>30</td>
<td>$3 \times 2$</td>
<td>4.96</td>
</tr>
</tbody>
</table>

Table 5.11: Results of the map image resolution algorithm.

Analysing the results of the Table 5.11, we conclude that except the service “SNIG CAOP Acores”, which returns a map image with a lot of white pixels, all the services returns map images for which it is possible to distinguish information in sub-images with $3 \times 2$ pixels.
5.5 Evaluation of the Services Testing

The objective of this experiment was to evaluate the GWD ability to let the GWD operator test geographic Web services.

The experiment consisted in using the GWD Playground component to configure and send requests to geographic Web services and to visualize the returned responses.

The services “CAOP Continente” and the “MassGIS”, both WMS implementations, were tested. Table 5.12 specifies the parameters of the requests sent to these services. The request sent to the “CAOP Continente” asks for the boundaries of the portuguese districts. The request sent to the “MassGIS” asks for the location of the Boston airports.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>CAOP Continente</th>
<th>MassGIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Width</td>
<td>400</td>
<td>800</td>
</tr>
<tr>
<td>Height</td>
<td>300</td>
<td>600</td>
</tr>
<tr>
<td>Layers</td>
<td>Distritos</td>
<td>massgis:GISDATA.OUTLINE25K_ARC, massgis:GISDATA.AIRPORTS_PT</td>
</tr>
<tr>
<td>Transparent</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>Format</td>
<td>image/png</td>
<td>image/png</td>
</tr>
<tr>
<td>SRS</td>
<td>EPSG:4326</td>
<td>EPSG:4326</td>
</tr>
<tr>
<td>BBOX</td>
<td>-10.1913, 36.8987, -5.7141, 42.1887</td>
<td>209899.0314465409, 874160.1289017625, 266889.786163522, 932123.0073665812</td>
</tr>
</tbody>
</table>

Table 5.12: Parameters of the “GetMap" requests.

In order to test the WFS provided by the service “Sigma”, a WFS implementation, the request to send was chosen from the list of example XML-requests provided by the GWD Playground. The selected request was the “Sigma_topStates.xml" that asks for a set of features representing some states of the United States of America.

In order to get the geographic coverages provided by the WCS of the National Snow and Ice Data Center (NSIDC), a “GetCapabilities" request was sent.

Ideally, in order to let the GWD operator assess by himself the geographic data quality, the responses returned by the services should be presented accordingly with their contents.

After sending the requests, we concluded that the requests were sent and the responses were presented, as expected, by the GWD application. Hence, the responses to the “GetMap” requests were presented over reference maps and through a map visualizer with controls to zoom, pan and switch between layers. The remaining responses, were presented as XML documents.

Figure 5.6 presents the “CAOP Continente" response, where is being showed the map of Portugal.
with the boundaries of districts at yellow.

**Figure 5.6**: Response returned by the service “SNIG CAOP Continente”.

Figure 5.7 presents the “MassGIS” response, where is being showed a map from Boston with the locations of the airports identified by plane symbols.

**Figure 5.7**: Response returned by the service “MassGIS WMS”.

Figure 5.8 presents the XML document returned by the “Sigma WFS”, where is being showed some feature coordinates of the West Virginia state.
Conclusions

This chapter described the experiments realized to evaluate the GWD application, which is an implementation of the framework proposed in the Chapter 3 for assessing geographic Web services. In this context, Table 5.13 presents a validation matrix for the requirements of the proposed framework, where it is possible to see the which are the requirements accomplished by each one of the GWD components. This validation matrix was filled based on the results achieved with the realized experiments.

Analysing the validation matrix, we can see that the requirement RD1, which refers to a plug-in for adding data quality algorithms to the GWD application, was not implemented. However, the GWD implements two data quality algorithms which can be applied to the data delivered by geographic Web services. The remaining requirements are accomplished by the GWD components ensuring the quality assessment of geographic Web services.

Moreover, the experiments presented in this chapter allow to conclude that the GWD application can be used to compare services provided by different providers. The conclusions of those experiments can be summarized as follows:

- The performance of the monitored services vary according to the size of their responses;
- The availability of the monitored services are close to the 99%;
- There are no significant differences in terms of scalability of the tested services provided by the SNIG and the IDEE;
- The WFS services provided by the IDEE, have feature coordinates with more resolution.
<table>
<thead>
<tr>
<th>Requirement</th>
<th>GWD Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>RA1 Registry services</td>
<td>✓</td>
</tr>
<tr>
<td>RA2 Query catalogues</td>
<td>✓</td>
</tr>
<tr>
<td>RM1 Monitor services</td>
<td>_</td>
</tr>
<tr>
<td>RM2 Configure monitoring function</td>
<td>_</td>
</tr>
<tr>
<td>RM3 Visualize monitoring function</td>
<td>_</td>
</tr>
<tr>
<td>RM4 Consult monitoring results</td>
<td>_</td>
</tr>
<tr>
<td>RM5 Compare monitoring results</td>
<td>_</td>
</tr>
<tr>
<td>RM6 Check responses consistency</td>
<td>_</td>
</tr>
<tr>
<td>RL1 Load-test services</td>
<td>_</td>
</tr>
<tr>
<td>RL2 Configure load-tests</td>
<td>_</td>
</tr>
<tr>
<td>RL3 Consult load-tests results</td>
<td>_</td>
</tr>
<tr>
<td>RD1 Add data quality algorithms</td>
<td>_</td>
</tr>
<tr>
<td>RD2 Apply data quality algorithms</td>
<td>_</td>
</tr>
<tr>
<td>RD3 Consult algorithm results</td>
<td>_</td>
</tr>
<tr>
<td>RP1 Configure requests to send</td>
<td>_</td>
</tr>
<tr>
<td>RP2 Invoke services</td>
<td>_</td>
</tr>
<tr>
<td>RP3 Visualize maps</td>
<td>_</td>
</tr>
<tr>
<td>RP4 Give example requests</td>
<td>_</td>
</tr>
<tr>
<td>RP5 Check responses consistency</td>
<td>_</td>
</tr>
</tbody>
</table>

Table 5.13: Validation matrix for the requirements of the framework proposed to assess geographic Web services.

- The WMS services available from the IDEE and the SNIG provide map image with equivalent resolution values.

than those provided by the SNIG;
Chapter 6

Conclusions and Future Work

This chapter summarizes the main findings of this work and discusses the future work regarding the different aspects of the research presented in this document.

6.1 Conclusions

This work proved that the quality of geographic Web services can be analysed from three different perspectives, namely 1) performance and availability monitoring, 2) scalability testing, 3) data quality assurance. A software framework, the GeoWatchDog (GWD), capable of supporting all the perspectives was designed and evaluated with services available on the Web.

The GWD has the following components:

- GWD Administrator;
- GWD Monitor;
- GWD Load-Tester;
- GWD Data-Tester;
- GWD Playground.

The GWD application was submitted to a set of experiments to prove its ability for assessing the quality of geographic Web services. The experiments consisted in using the GWD components to evaluate each function of the GWD application. These experiments proof that the GWD application provides quality estimates to potential users or owners of geographic services, helping to determine if the service is suitable for their needs. The GWD application provides the following functionalities:
• Administration of the GWD system, by adding, editing and deleting services;
• Configuration of the performance and availability monitoring;
• Configuration of test-plans for the scalability testing;
• Assessment of the geospatial data quality;
• Testing operations for experimenting and invoking the services by editing or creating requests.

Furthermore, the GWD application provides interfaces (such as graphs and tables) to visualize and analysis the results of its components.

Since the Open Geospatial Consortium (OGC) standards has an important role in developing and publishing geographic Web services, the GWD application supports various of these standards. Furthermore, it is possible to extend the support for more standards.

In terms of the GWD application development, the owsWatch had an important role as the basis for the GWD architecture. Moreover, the monitoring function of the owsWatch was used in the GWD application.

The GWD architecture implements the MVC design pattern for the developing of Web-based applications, where the model is a set of files, which hold the essential information to run the GWD application. The controller implements the GWD behavior. The view is a set of interfaces that allow the interaction between the human operator and the GWD application.

6.2 Future Work

One of the identified requirements in 3.4 for the proposed framework was a plugin for adding quality assessment algorithms for spatial data. This requirement was not accomplished, although it would be highly useful to implement the plugin in the GWD application. For implementing this requirement, the GWD application could provide an interface to define the algorithm by specifying parameters such as:

• Algorithm name;
• Algorithm description;
• Service type for which the algorithm applies;
• Service request type for which the algorithm applies;
• Algorithm result parameters;
• Java class that implements the algorithm.
The information above could be stored in a XML file. Figure 6.1 presents a possible XML file describing an algorithm. The presented file specifies the “Map Image Resolution” algorithm, which measures the maximum resolution of map images returned by a WMS in response to a “GetMap” request. The results of this algorithm are the bounding box and the layer specified by the “GetMap” request, and the measured resolution.

```
<Algorithm>
  <Name>Map Image Resolution</Name>
  <Description>Measure the max resolution of map images</Description>
  <Service>WMS</Service>
  <Request>GetMap</Request>
  <ResultParams>
    <BBOX unit=degrees/>
    <Layer unit=pixels/>
    <MaxResolution unit=km/>
  </ResultParams>
  <Class>c:/qualityAssessmentAlgorithms/MapImageResolution.class</Class>
</Algorithm>
```

**Figure 6.1:** Sample of an algorithms file.

Section 3.5 identified the need to present service responses over reference maps (requirement RP3). Currently, the GWD Playground component allows to visualize maps returned by a WMS over reference maps. This is done through the WMS Openlayers client. Openlayers also provides a WFS client which could be used in the GWD application for visualizing features returned by a WFS over reference maps.

SEXTANTE[40] is a free geoprocessing library for the development of algorithms focused in geospatial analysis. Using the Sextante we can develop, implement and use the algorithms through graphical interfaces such as a toolbox and a graphical modeler. Moreover, SEXTANTE can be integrated with other applications through the using of Java libraries. Sextante is being developed by the “Junta de Extremadura” (local government of Extremadura, Spain). As a suggestion for a future work, the algorithms implemented in the SEXTANTE could be used by the GWD Data-Tester component.
References


