U1051 Unigis MSc 2003
DI Michael Hadrbolec

Web Map Context Documents

Reality check of the specification

16th of March, 2005

Erklärung

Ich versichere, diese Master Thesis ohne fremde Hilfe und ohne Verwendung anderer als der angeführten Quellen angefertigt zu haben, und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat. Alle Ausführungen der Arbeit, die wörtlich oder sinngemäß übernommen wurden, sind entsprechend gekennzeichnet.

Abstract

This thesis answers the question, whether the OGC "Web Map Context Documents" specification is ready to be used in a complex real world scenario. Based on the requirements of a groundwater expert, who has to answer questions, prepares reports and assists decision makers in authorities the required GIS functionality is derived. First these functions are checked in theory, and it is assesed if they are covered by the specification. In the next step they are implemented. Finally, a practical test is carried out as "proof of concept" with the OGC OWS reference WMS and WFS implementations.

Zusammenfassung

Diese Arbeit beschäftigt sich mit der Fragestellung, ob die OGC "Web Map Context Documents" Spezifikation für ein reales, komplexes Szenario geeignet ist. Der Ausgangspunkt sind die Anforderungen eines Grundwasser Experten, der Fragen beantwortet, Berichte erstellt und für Entscheidungsträger in Behörden den fachlichen Input liefert. Aus den sich daraus ergebenden Anforderungen wird die benötigte GIS - Funktionalität abgeleitet. Am Beginn steht eine theoretische Untersuchung, ob die Spezifikation die notwendigen Funktionen unterstützt. Der nächste Schritt ist die Implementierung ebendieser. Abschließend erfolgt als „Proof of Concept" der praktische Test der identifizierten GIS Funktionen gegen die OGC OWS Referenz-implementation für WMS and WFS.

Danksagung

Mein Dank gilt Dir Kerstin, für Deine Liebe, Dein Einfühlungsvermögen, Deine Geduld und Aufmunterung, wenn es „mal wieder nicht so lief“ und den konstruktiven Input. Ohne Dich wäre es nicht möglich gewesen.

Meinen Eltern, die in mir Neugierde und Lust zu Lernen gefördert haben.

Weiter möchte ich mich beim UNIGIS Team für ihre Hilfe während des Fernstudiums bedanken. Macht weiter so!

Table of contents

Figures

Tables

1 Preface

1.1 Service orientated architecture (SOA)

Service orientated architecture (SOA) is at present the “catchword” in the information technology (IT) industry.

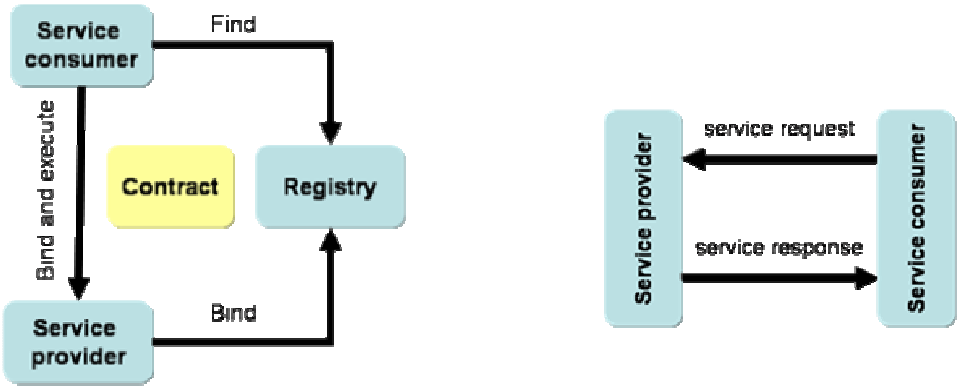


Figure 1 Service orientated architecture (SOA) (Mc Govern et al., 2003)

The SOA ”find-bind-execute” paradigm as shown in the figure above allows the consumer to query a registry for the service that meets his requirements. If a matching service is found the registry returns a contract, which specifies how to interact with the service and an endpoint (the physical address where the service runs) for the service. Once found, the service is used from the returned endpoint according to the specified contract. A service can be described as a black box, which encapsulates function(s) and does not depend on the context or state of another service.

Characteristics of SOA according Mc Govern (2003, pp. 43)

- *Services are discoverable and dynamically bound*
- *Services are self-contained and modular*
- *Services stress interoperability*
- *Services are loosely coupled*
- *Services have network addressable interfaces*
- *Services have coarse-grained interfaces*
- *Services are location transparent*
- *Services are composeable*

This emerging trend has also reached the geographic information science. It is used inside several specifications of the **O**pen **G**eospatial **C**onsortium (OGC).

1.2 Problem definition

Usually, services of different service providers, such as e.g. Web Mapping Services (WMS), Web Feature Services (WFS), and Web Coverage Services (WCS) are used and accessed simultaneously.

E.g. a base map with the administrative boundaries, groundwater bodies, sampling sites, etc.

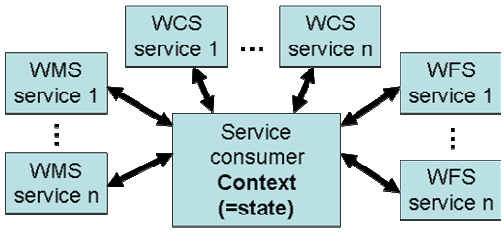


Figure 2 Simultaneous use of multiple services

As mentioned in the preface, the SOA services do not depend on the state (=context) of other services. The **problem** for a service consumer is to **hold and store** the **context of multiple, distributed services independent of a particular client,** which is in this case an OGC service.

1.3 Promising techniques

In June 2003 the "Web Map Context Documents" Version 1.0.0 (WMC) specification was published.

"The present Context specification states how a specific grouping of one or more maps from one or more map servers can be described in a portable, platform-independent format for storage in a repository or for transmission between clients. This description is known as a "Web Map Context Document," or simply a "Context." Presently, context documents are primarily designed for WMS bindings. However, extensibility is envisioned for binding to other services. A Context document includes information about the server(s) providing layer(s) in the overall map, the bounding box and map projection shared by all the maps, sufficient operational metadata for Client software to reproduce the map, and ancillary metadata used to annotate or describe the maps and their provenance for the benefit of human viewers." (Humblet, 2003 pp. viii)

It defines how to **store** persistent reusable **information** about the **context of distributed web mapping services** in a **platform-independent** manner. This specification promises to solve the obstacle mentioned above in the problem definition.

1.4 Aim of the thesis

The **aim** of this thesis is to **critically review** the "**Web Map Context Documents**" **specification** (WMC) of the Open Geospatial Consortium (OGC) and to assess **whether it can be used under real world conditions or not**.

1.5 Description of the approach

Specifications are necessary to foster interoperability and standards, but they are only relevant and can gain broader acceptance if they meet the requirements of the user. Therefore, the **W**eb **M**ap **C**ontext document specification (WMC) can be best evaluated with the help of one or more real world examples.

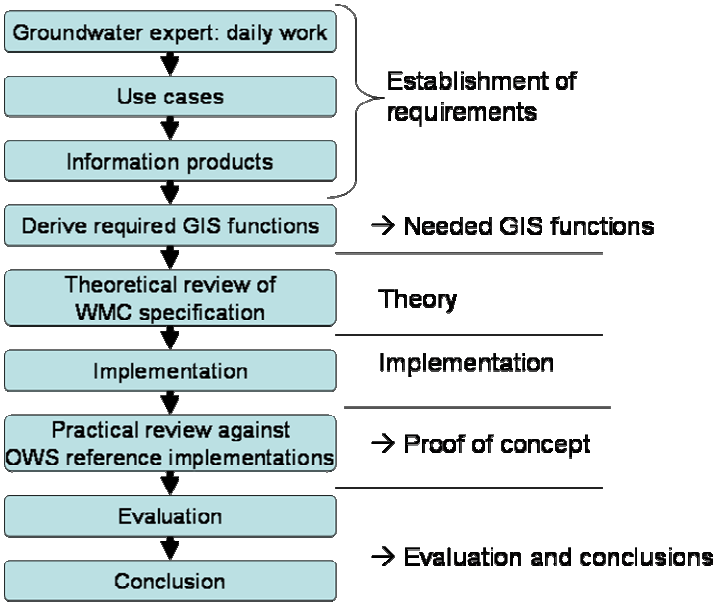


Figure 3 Overview of the approach

1.5.1 Establishment of requirements

The starting point is a real world scenario based on the daily work of a ground water expert. The following **three use cases**, which cover major parts of his work, are reviewed in detail:

1. Get an overview of the situation (**Browse**) → 2.3.1 Use case: "Browse"
2. Compile a report (**Report**) → 2.3.2 Use case: "Report"
3. Perform explorative analysis (**Explore**) → 2.3.3 Use case: "Explore" (explorative analysis)

The author follows Tomlinson's methodology (2003) to identify the information products (IP) for theses use cases. An information product is the desired output from a **G**eographic **I**nformation **S**ystem (GIS). This can be a map, report, list, graph and any

combination of these elements. The IPs are analysed to identify the steps to generate them. The result will be the **list** of the required **GIS functions**.

1.5.2 Theoretical review

In a next step the "Web Map Context Documents" specification is evaluated in view of each of the identified **functions** and its theoretical coverage.

1.5.3 Implementation

As prerequisite for the practical review it is necessary to implement the specification. Utilities will be developed to perform the necessary GIS functions e.g. scale, move, identify, style, "GetMap" WMS request, etc. .

1.5.4 Practical review (proof of concept)

According to the Cite Homepage (2004) the software products "Deegree" and "Geoserver" are the reference implementations of Web Mapping Service (WMS) and Web Feature Service (WFS). These two reference implementations will be used, to **check practically if they are able to perform** the GIS **functions** implemented in the previous step.

1.5.5 Evaluation and conclusions

The outcome will be evaluated and the lessons learned from these steps will be discussed. The work will finally conclude whether the **specification meets the identified criteria.**

2 Establishment of requirements

2.1 The big picture

It is state of the art in the **I**nformation **T**echnology (IT) industry to model a real world scenario with use case diagrams to get a better overview.

"A use case […] is a set of scenarios tied together by a common user goal." Fowler (2000 p. 40).

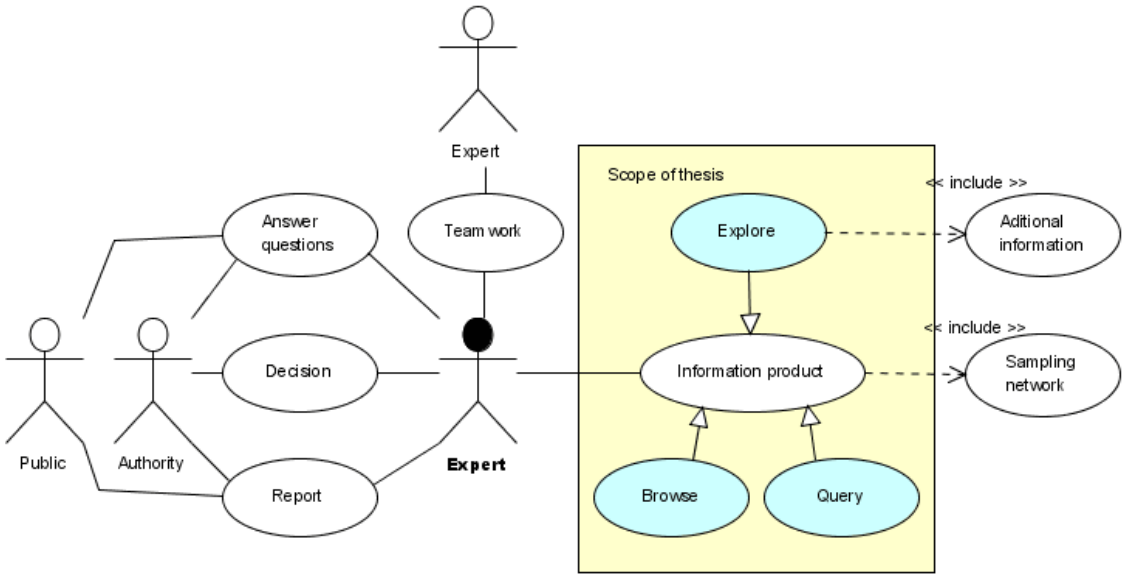


Figure 4 Groundwater expert use cases diagram

The groundwater expert has to answer questions / requests from the general public and administration about the situation of groundwater bodies and their sampling sites. He provides in close cooperation with the responsible authorities decision support for countermeasures if critical limits are exceeded. In regular intervals the expert prepares reports about the situation and future trends for involved authorities and the general public. In order to fulfil these requirements the expert needs adequate information products. This includes primarily information on the state of the groundwater bodies and the measured parameters of the sampling sites. In some cases external information like e.g. contaminated sites is needed to answer hypotheses about anomalies if critical limits are exceeded at specific sampling sites. Additionally, it should be also possible to exchange information (teamwork) with other experts.

2.1.1 Stakeholders

A stakeholder is an organisation, group and or person with an interest in the process either now or in the future.

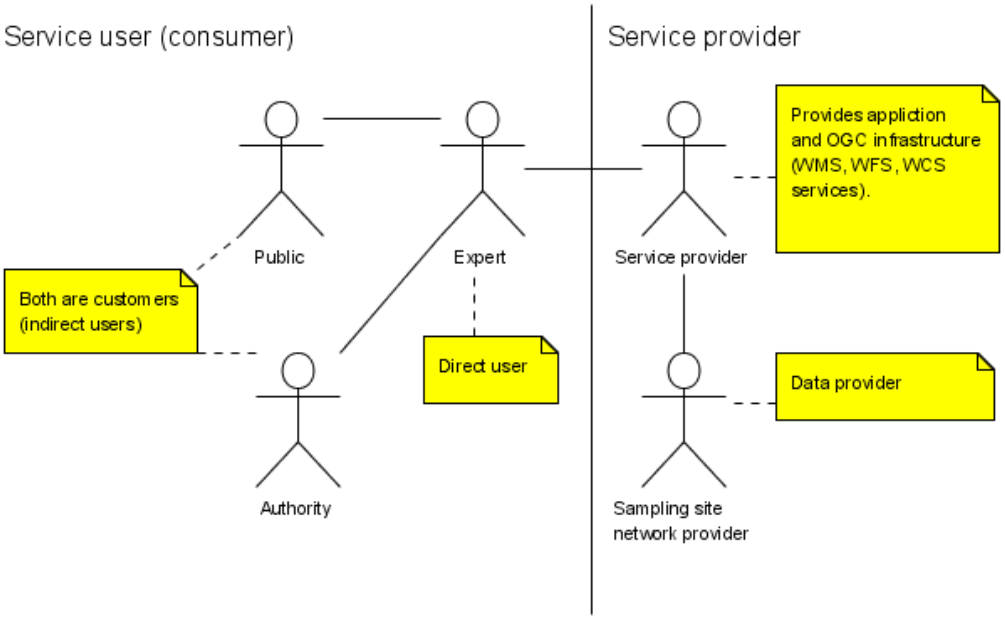


Figure 5 Stakeholders and their mutual relations

Based on the use case diagram above the following two stakeholder groups, their interests and goals are identified:

1. Service provider
2. Service user

2.1.1.1 Service provider

A service provider is an actor who provides either data and / or a service.

Stakeholder	Interest and goal
Sampling site network provider (Provincial authorities)	These authorities have legal obligations to monitor groundwater bodies. Their goal is a better access to information retrieved from their datasets. They need homogeneous information products across provincial borders.
Service provider (Umweltbundesamt)	The agency has legal information obligations towards the public and experts. The organisation has budget restrictions and wants to minimize phone, e-mail, fax and written requests. For this reason a better and more cost efficient automated interactive information service is needed.

Table 1 Service provider

2.1.1.2 Service user (consumer)

A service user is an actor that consumes (uses) such services and / or their interpretations.

Stakeholder	Interest and goal
Groundwater experts (Local, provincial and federal level, civil engineering companies)	They need to access the information for their daily work (reports, decision support, and public information) in a user-friendly, intuitive way.
Authority (Local, provincial, federal level)	An authority wants interpreted information (reports) from the experts. They need fast and accurate decision support.
Public	The public wants easily understandable information when they need it.

Table 2 Service user (consumer)

2.2 Scope

The scope of the reviewed use cases is limited to information products required by the groundwater expert related to the "Web Map Context Documents" (WMC) specification. Only the specification is reviewed. Proprietary extension mechanisms like the "Extension" element go beyond the specification, break interoperability and are skipped for that reason. The above mentioned use cases are evaluated in a greater detail according to Tomlinson (2003). They are needed to define the required **I**nformation **P**roducts (IP). The description of an IP is compiled of:

- map requirements,
- list or report requirements,
- document and image requirements and
- the steps to generate the products including the required GIS functions.

The Appendix "Glossary of GIS functions" (pp. 127) contains a detailed description of the mentioned GIS functions.

2.3 Detailed use cases description

2.3.1 Use case: "Browse"

The starting point for every work like a report or an assessment is, to obtain an overview of the groundwater body. The shape of the body, the distribution of the

sampling sites, its surroundings and other specific features will be taken into account. The goal is to get an impression of the special situation of a specific body. This means that the experts need a possibility to look at the area and its surroundings at different scales and extents. In addition, it is necessary to retrieve interactively further information (e.g. like the identify tool in standard GIS software).

2.3.1.1 Scenario / Steps

The user selects an area of interest (e.g. a specific groundwater body). A map is generated in which the sampling sites are visualised according to their measured parameters. Depending on the scale more or less details are shown. The expert scales (zooms in and out), moves and pans the area of interest. On mouse click background information about the selected feature is provided.

2.3.1.2 Information product

2.3.1.2.1 Map requirements

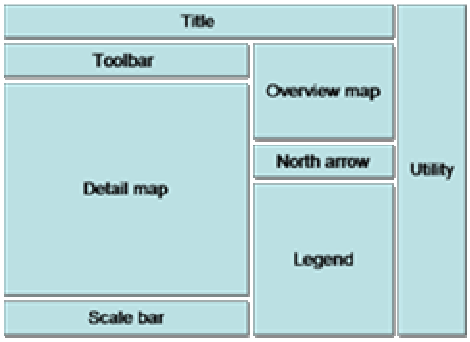


Figure 6 Map requirements: browse use case

The map has a title including the logo of the service provider. The detailed map shows the current extent and allows user interaction. A small scale overview map is necessary in which the extent of the detailed map is displayed as rectangle. A north arrow, a scale bar and a legend are provided. The interaction mode is set via the toolbar which toggles between different map interaction modi (e.g. like scale, pan, identify, search, etc.). Expert GIS functions like geocoding are invoked with wizard based utilities. It is necessary to provide a tool for measuring the distance between features.

2.3.1.2.2 List or report requirements

Only very simple structured lists are necessary. They are needed for identification of the features during the browsing process.

2.3.1.2.3 Document and image requirements

None

2.3.1.2.4 Steps to create the product

Data needed	Function
Administrative boundaries, Groundwater bodies, Sampling sites, Rivers, Lakes	ADDRESS GEOCODE (search): e.g. Highlights and jumps to an area of interest (usually an administrative unit, a groundwater body or a sampling site).
	SYMBOLIZE (predefined): e.g. Displays sampling sites according to their measured values.
	GRAPHICAL OVERPLOT: e.g. Graphically overlays sampling sites, groundwater bodies, rivers, lakes and administrative boundaries.
	BROWSE: Moves the displayed map extent and allows to identify features.
	SCALE CHANGE: Changes the scale of the displayed map, e.g. zoom in and out.
	MEASURE LENGTH: e.g. Measures the distance from a sampling site to a point of interest.

Table 3 Steps to create the "Browse" information product

2.3.2 Use case: "Report"

The groundwater expert periodically aggregates a report about the groundwater bodies and sampling sites according to predefined reporting criteria.

2.3.2.1 Scenario / Steps

First a groundwater body is selected. Afterwards the expert browses the groundwater body and its surroundings to get a feeling about the whole situation. Unnecessary information is suppressed. Then the expert creates multiple maps, graphs and lists according to the report requirements by performing spatial and attributive queries. The results are classified and symbolised according to the reporting conventions. It must be possible for the expert to display each result on the screen as well as to plot it in a specific scale. Furthermore, it has to be possible to store the output in a digital format for further processing (e.g. with statistical, word-, and image-processing software, etc.).

2.3.2.1.1 Use case extensions

- Save (serialize / store) the current state of the work as WMC document and access (load / deserialize) it later on.
- Share / exchange the work as WMC document with other experts.
- Annotate and highlight specific features.

2.3.2.2 Information product

All requirements of the “Browse” (p. 21) use case apply. Only additional requirements are mentioned here.

2.3.2.2.1 Map requirements

The toolbar and the utilities have to be enhanced to allow the execution of the supplementary functions (query, classify, export, save, load, annotate, exchange, etc.).

2.3.2.2.2 List or report requirements

According to the respective reporting requirements various lists are necessary.
E.g.: Report of each sampling site of a groundwater body for each measured parameter and a given time interval.

2.3.2.2.3 Document and image requirements

The report list has to be provided in a format which allows further processing in text-processing and / or spreadsheet software. Maps and graphs have to be provided in a way that allows to include them in a report without any further manipulation. The symbolisation has to conform to the respective reporting standard, which can vary between different kinds of reports.

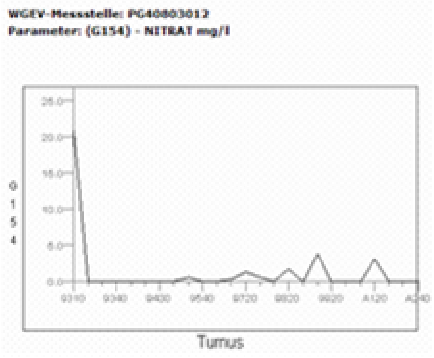


Figure 7 Example: time series graph

2.3.2.2.4 Steps to create the product

In addition to all steps and datasets of the "Browse" use case the following requirements apply:

Data needed	Function
All datasets from the previous use case	All functions described in the "Browse" use case.
	SURPRESS: e.g. Surpresses the sampling sites outside the area of interest.
	SPATIAL QUERY: Creates a subset based on geometry (e.g. All sampling sites within a groundwater body).
	ATTRIBUTE QUERY: Creates a subset based on attribute values (e.g. All sampling sites that exceed a limit).
	GENERATE BUFFER: e.g. Creates a buffer along rivers.
	CLASSIFY ATTRIBUTES: e.g. Classifies sampling sites according to their measured values.
	CLIP: e.g. Extracts sampling sites within an administrative boundary.
	EDIT AND DISPLAY (on output): e.g. Prepares the further output to paper or file.
	SYMBOLIZE: e.g. Symbolises according to the report criteria (predefined). Annotate highlight (user defined)
	GENERATE LIST (REPORT): e.g. Creates a list of a spatial query.
	GENERATE GRAPH: e.g. Creates a graph for the parameter along the time axis.
	PLOT: e.g. Generates a hardcopy output of map, graph and lists.
WMC document	SERIALIZE (save / store): e.g. Saves the current WMC document.
	DESERIALIZE (load): e.g. Loads the serialised WMC document.
	EXCHANGE information (WMC document): e.g. Share a WMC document between other experts.

Table 4 Steps to create the "Report" information product

2.3.3 Use case: “Explore” (explorative analysis)

During the previous two use cases the expert may have identified anomalies which need further investigation and / or explanation. The expert develops a hypothesis and tries to validate it with additional layers (datasets).
E.g.: A measured value of a sampling site exceeds the critical limit. The expert has the hypothesis that a potential contaminated site could be the cause. Therefore, he adds the layer and validates his hypothesis.

2.3.3.1 Scenario / Steps

In addition to the reporting use case it is necessary to add one or more additional layers (datasets). Query, browse, scale, symbolise, etc. functions are carried out like in the previous “Report” and “Browse” use cases. These steps are repeated as often as necessary. This leads either to a rejection or verification of the hypothesis.

2.3.3.2 Information product

All requirements of the “Report” (p. 23) use case apply. As in the “Report” use case only the differences are mentioned here by the author.

2.3.3.2.1 Map requirements

The map requirements are the same as for the “Report” use case. Only a possibility to add datasets on the fly has to be included.

2.3.3.2.2 List or report requirements

The same requirements as for the “Report” use case apply here.

2.3.3.2.3 Document and image requirements

Again the same requirements as for the “Report” use case apply here.

2.3.3.2.4 Steps to create the product

In addition to all steps and datasets of the “Report” use case the following requirements apply:

Data needed	Function
All datasets from the “Report” use case	All functions described in the “Report” use case.
The respective layer which should be added. It has to be provided as Open Geospatial Consortium Open Web Service (OGC OWS) which can be either a Web Feature Service (WFS) or a Web Coverage Service (WCS).	ADD: Adds a new layer from an OWS service
	PROJECTION CHANGE: Changes the projection of the OWS, if necessary
	CLIP: Clips the OWS service to the area of interest.
	SYMBOLIZE: Symbolises the OWS service according user preferences
	GRAPHICAL OVERLAY: Graphically overlays the OWS service with other layers, to obtain an overview of the dataset.
	BROWSE: Browses OWS services.
	SPATIAL QUERY: Creates a subset of the OWS service based on geometry.
	ATTRIBUTE QUERY: Creates a subset of the OWS service based on attribute values.
	GENERATE BUFFER: e.g. Creates a buffer along a rivers
	CLASSIFY ATTRIBUTES: Classifies the OWS service.
	SURPRESS: Suppresses parts of the OWS service.
	GENERATE LIST (REPORT): Creates a list for the OWS service.
	GENERATE GRAPH: Creates a graph for the parameter of the OWS service.

Table 5 Steps to create the product

2.4 Functions and their estimated frequency of use

Based on the three use cases described above the following list of required GIS functions is compiled. They are grouped according Tomlinson (2003, Lexicon pp. 255-276). In order to evaluate the importance of the different functions, a rough estimation of the frequency of use, based on expert judgement, has been carried out. The system functions are grouped into the following two categories:

- Basic system capabilities
- Data manipulation and analysis functions

2.4.1.1 Basic system capabilities

The basic system capabilities are needed to get data into a geographic information system (GIS), store it, manage it and output it.

Function	Browse	Report	Explore
Data input			
Load WMC file		x	x
Meta-information WMC file		x	x
Exchange WMC file (Teamwork)		x	x
Storing maintaining and outputting data			
Serialize WMC file for later use (Store)		x	x
Edit and display (on output)			
Visibility			
Display		x	x
Hide		x	x
Add / remove layer			
Add layer			
Add predefined layer		x	x
Add user defined layer			x
Remove layer		x	x
Order layer		x	x
Symbolize			
Predefined	x	x	x
User defined			x
Annotate highlight		x	x
Plot		x	x
Browse			

Move	xxx	xxx	xxx
Function	**Browse**	**Report**	**Explore**
Identify	xx	xx	xx
Suppress		x	x
Create list (report)		xx	xx

Frequency of use: x…rarely, xx…often, xxx…very often

Table 6 Basic system capabilities

2.4.1.2 Data manipulation and analysis functions

These functions are used to manipulate data in the GIS in preparation for analysis, and to generate new data and information through analysis and modelling.

Function	Browse	Report	Explore
Query			
Spatial query		xx	xx
Attribute query		xx	xx
Combined query		x	x
Generating features, views, and graphs			
Generate buffer		x	x
Generate graph		xx	xx
Manipulate features			
Classify attributes		xx	xx
Clip		x	x
Scale change	xxx	xxx	xxx
Projection change			x
Address location			
Address geocode	x	x	x
Measurement			
Measure length	x	x	x
Spatial analysis			
Graphical overplot	xxx	xxx	xxx

Frequency of use: x…rarely, xx…often, xxx…very often

Table 7 Data manipulation and analysis functions

3 Theoretical review

According to the author's approach the next step is to examine the required GIS functions which were identified in the previous step. One after the other is reviewed in detail regarding their coverage by the "Web Map Context Documents" specification. The level of detail of the review depends on the estimated frequency of use of the respective GIS function (Table 7, p. 29).

3.1 Specification overview

Basically, the specification addresses only WMS services. *"This specification is relevant to clients of the OGC Web Map Service [...]."* Humblet (2003 pp. 9)
The WMS 1.1.1 specification (Beaujardiere, 2002) supports as an optional feature the "Styled Layer Descriptor Implementation Specification" (SLD). Besides its capability for user-defined map styling, a SLD enabled WMS server is able to integrate other OGC "Open Web Service"'s (OWS) like "Web Feature Services" (WFS) and / or "Web Coverage Services" (WCS). They run either inside the same server as "integrated server" or on a different machine as "component server" (Lalonde 2002, pp. 6). With this mechanism it is possible to integrate WFS and WCS as well. The "Web Map Context Document" specification describes two different document types:

- "ViewContextCollection" and
- "ViewContext" documents.

3.1.1 ViewContextCollection document

A "ViewContextCollection" document stores links to one or more "ViewContext" documents. It is used to handle multiple views (see "Handling multiple views" p. 39 for further details).

3.1.2 ViewContext document

The "ViewContext" document stores all relevant information which is needed for displaying and interacting with a certain view (context). It contains e.g. the displayed extent, the layers and their styles, whether or not they can be queried. It can be compared with a project file of a traditional GIS-software.

3.2 Basic system capabilities

3.2.1 Data input

3.2.1.1 Load WMC file (deserialize)

Load is the process of retrieving information of a persistent (stored / saved) document. It is necessary to be able to load a “ViewContext” and or “ViewContextCollection” document from either the file system or an “Uniform Resource Locator” (URL). After loading, the document has to be parsed for further automated processing.

3.2.1.2 Meta-information of a WMC file

Meta-information is basically “data about data”. It aims to *“[…] provide a clear procedure for the description of digital geospatial datasets so that users will be able to locate geographic data, to determine whether the data in a holding will be of use to them, and how to access the data.”* Danko (2005)

The review of meta-information describes separately the two different document types:

- “ViewContextCollection” and
- “ViewContext” documents.

3.2.1.2.1 “ViewContextCollection” specific meta-information

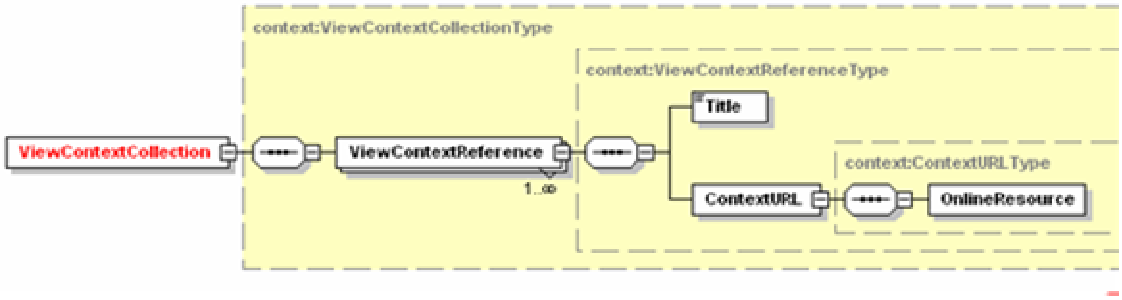


Figure 8 “ViewContextCollection” XML schema overview

The “**ViewContextCollection**” contains one ore more “**ViewContextReference’**s”. Each of them is identified with a unique “**id**” attribute. The schema attributes are skipped for better readability in all following as figure displayed XML schemas. The “**Title**” is a human readable description for the "**OnlineResource**" referenced in the “**ContextURL**”, which stores the URL of the respective “ViewContext” document. Unfortunately, additional meta-information tags for describing the collection as a whole are not available. E.g. it is not possible to specify a contact person or an abstract or further meta-information for the ViewContextCollection.

```
<?xml version="1.0" encoding="UTF-8"?>
<ViewContextCollection xmlns="http://www.opengis.net/context" version="1.0.0">
  <ViewContextReference version="1.0.0" id="MyCollection">
    <Title>WebMapContextDocument</Title>
    <ContextURL>
      <OnlineResource
        xlink:href="http://127.0.0.1:8084/deegreewms/debug/wmc/RemoteWFS_States.xml"/>
    </ContextURL>
  </ViewContextReference>
  … all other  "ViewContextReference" elements …
</ViewContextCollection>
```

Figure 9 "ViewContextCollection" document fragment example

3.2.1.2.2 "ViewContext" specific meta-information

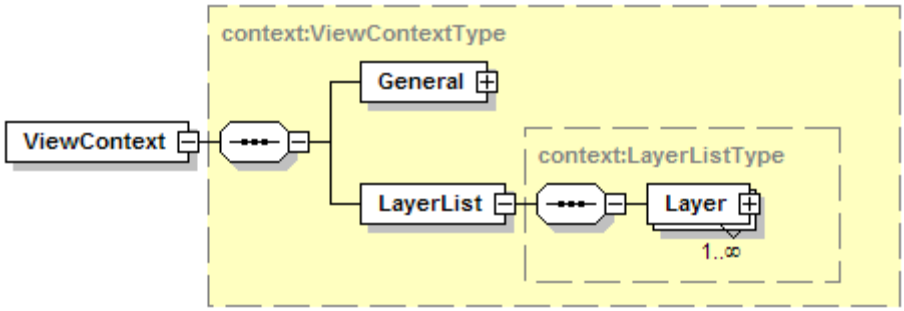


Figure 10 "ViewContext" XML schema overview

According to Humblet (2003) a "ViewContext" document is divided in a "**General**" **section**, which represents information relevant to the whole view (context) and a **layer specific part**, which is only relevant for a specific layer. Due to this fact meta-information is available on the following different levels:

- **General meta-information** and
- **layer specific meta-information**.

3.2.1.2.2.1 General meta-information

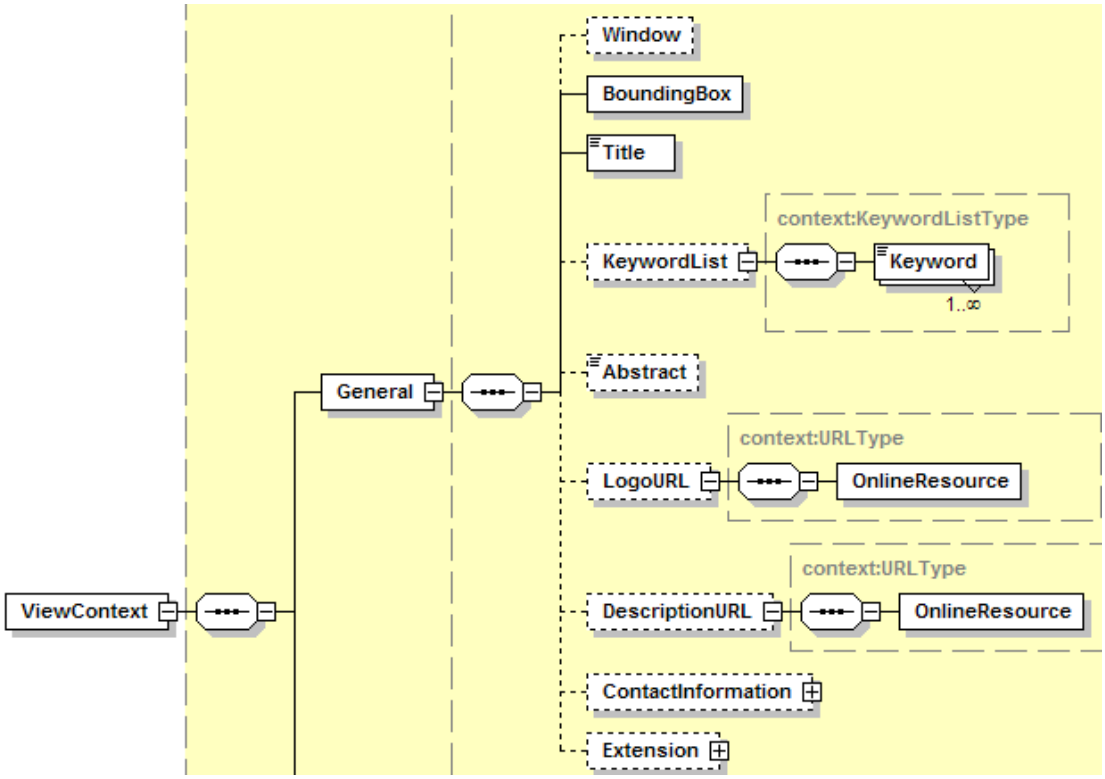


Figure 11 "ViewContext" XML schema: General meta-information elements

Each "**ViewContext**" has a unique identifier which is specified by the required "**id**" attribute. "**General**" identifies the section where the whole context relevant information is stored. This element is required. The optional "**Window**" specifies the width and height of the "ViewContext" in pixels. The required "**BoundingBox**" is specified as defined in the WMS 1.1.1 specification (Humblet, 2003) and contains the spatial reference system ("**SRS**") and the extent (**minx**, **miny**, **maxx, maxy**) of the displayed view in map units. Each ViewContext should have a human readable title ("**Title**"). This may be used to display a map title. Keywords should allow to search within the document ("**KeyWordList**", "**KeyWord**"). It is questionable if users should provide keywords, which will only work if the input is standardised. The tendency in most businesses is to index the whole document and use a thesaurus in combination with ontologies to perform a full text search. This is also possible, because XML documents like the ViewContext are simple text documents. The optional "**Abstract**" element describes the content of the context document as text. Sometimes the logo of an institution or project has to be included. In this case, the optional "**LogoUrl**" element is used. An external resource can be referenced with the optional "**DescriptionURL**" element. This can be used to provide extensive documentation and explanations.

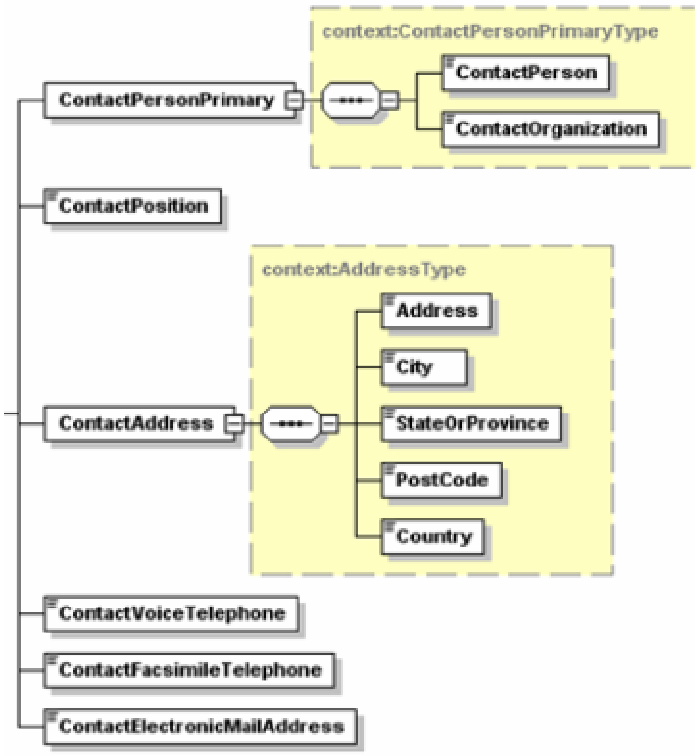


Figure 12 "ViewContext" XML schema: "ContactInformation" element

The optional "**ContactInformation**" represents the same information as specified in WMS 1.1.1 and / or in the WMS 1.3.0 specification. With this information it is possible to communicate with the person who has created the respective view context.

Proprietary extensions of GIS-software providers are made inside the "**Extension**" element.

```
<ViewContext version="1.0.0" id="U1051">
   <General>
      <Window width="400" height="200"/>
      <BoundingBox SRS="EPSG:4326" minx="-0.005" miny="-0.005" maxx="0.005"
maxy="0.005"/>
      <Title>UniGIS MSc 2003 master thesis</Title>
      <KeywordList>
         <Keyword>Master thesis</Keyword>
         <Keyword>U1051</Keyword>
      </KeywordList>
      <Abstract>
         Master thesis MSc UniGIS MSc 2003. This view context document stores the examples used
         within the master thesis.
      </Abstract>
      <LogoURL format="image/gif" width="126" height="80" >
          <OnlineResource xlink:type="simple"
              xlink:href="http://127.0.0.1:8084/deegreewms/images/myLogo.gif"/>
      </LogoURL>
      <DescriptionURL>
```

```
        <OnlineResource xlink:type="simple"
                    xlink:href="http://127.0.0.1:8084/description/Description.html"/>
      </DescriptionURL>
      <ContactInformation>
        <ContactPersonPrimary>
          <ContactPerson>Michael Hadrbolec</ContactPerson>
          <ContactOrganization>Agency</ContactOrganization>
        </ContactPersonPrimary>
        <ContactPosition/>
        <ContactAddress>
          <Address>Elmsstreet 13</Address>
          <City>Vienna</City>
          <StateOrProvince>Vienna</StateOrProvince>
          <PostCode>0815<PostCode>
          <Country>Austria</Country>
        </ContactAddress>
        <ContactVoiceTelephone>0815</ContactVoiceTelephone>
        <ContactFacsimileTelephone>0915</ContactFacsimileTelephone>
        <ContactElectronicMailAddress>test@test.net</ContactElectronicMailAddress>
      </ContactInformation>
    </General>
      … The rest of the viewCotext document ….
</ViewContext>
```

Figure 13 example "ViewContext " document fragment : General meta-information

The main gap of the general meta-information is that it is only allowed to provide one "DescriptionURL", "Logo" etc.

3.2.1.2.2.2 Layer-specific Meta-information

Every layer can reference a different data source. Therefore, it is necessary to specify the service ("**Server**" element) from which the layer is retrieved. This is done by the "**Service**" attribute which contains the "**Service**" and "**Version**" attributes. They determine the type (OGC:WMS, OGC:WFS or OGC:WCS) of the service. The "**OnlineResource**" points to the physical address where the service runs.

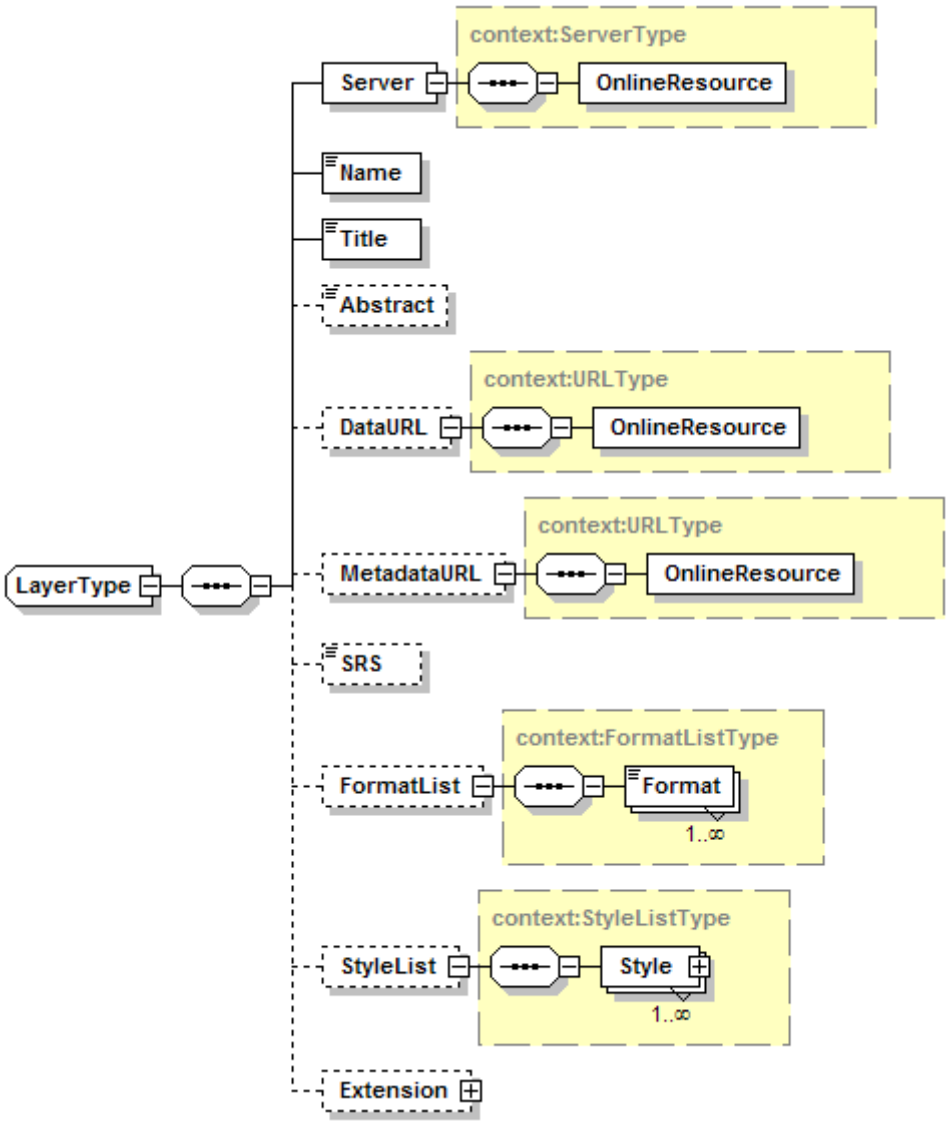


Figure 14 "ViewContext" XML schema: Layer specific meta-information elements

Each layer needs a unique id which is derived from the "GetCapabilities" request and is stored inside the required "**Name**" element. The required "**Title**" is once more the human readable label of the layer. A textual description of the layer can be provided with the optional "**Abstract**" element. Also optional is the "**DataURL**" element which contains a link to an online resource where information about the layer can be found. An optional "**MetadataURL**" element contains a link to an "**OnlineResource**", where additional descriptive layer specific meta-information can be found. The optional "**SRS**" element should store a list of supported spatial reference systems (SRS) for this layer. A map can only be displayed in those SRS that are supported by all displayed layers of a view context. According to the specification each layer has one or more SRS. A closer look at the specifications XML schema shows an error. It can store only one SRS. All possible map image output formats are included under the "**FormatList**" element. At least one "**Format**" has to be provided. The formats are expressed in the

same way as in the WMS 1.1.1 specification. The active format is marked by setting the "**current**" attribute to "true". Each layer can be visualised in multiple ways. This different styles ("**Style**") are stored in the "**StyleList**" element (see "Symbolize" p. 42 for further details). At least one "Style" has to be provided per layer.

```
<ViewContext version="1.0.0" id="">
  <General>
      … see Figure 13  example "ViewContext " document fragment : General meta-information ….
  </General>
  <LayerList>
    <Layer>
      <Server service="OGC:WMS" version="1.1.1">
        <OnlineResource xlink:type="simple"
            xlink:href="http://127.0.0.1:8084/deegreewms/wms?"/>
      </Server>
      <Name>testlayer</Name>
      <Title>Test layer</Title>
      <Abstract>
        Master thesis MSc UniGIS MSc 2003. This view context document stores the examples
        used within the master thesis.
      </Abstract>
      <MetadataURL>
        <OnlineResource xlink:type="simple"
            xlink:href="http://127.0.0.1:8084/deegreewms/wms?"/>
      </MetadataURL>
      <DataURL>
        <OnlineResource xlink:type="simple"
            xlink:href="http://127.0.0.1:8084/deegreewms/wms?"/>
      </DataURL>
      <SRS>EPSG:4326</SRS>
      <FormatList>
        <Format>image/png</Format>
        <Format current="true">image/gif</Format>
        <Format>image/jpeg</Format>
      </FormatList>
      <StyleList>
        <Style current="true">
          <Name>default </Name>
          <Title>Default Style for Test layer</Title>
        </Style>
          … The other Style's for this Layer ….
      </StyleList>
    </Layer>
          … The other Layers ….
  </LayerList>
</ViewContext>
```

Figure 15 Example "FormatList" element

The main limitation of the WMC specification is that there is no possibility to store information about the layer attribute names and data types (geometry [raster / vector, point, polyline, polygon], character, double, integer, etc.). This causes troubles if an analysis has to be performed (e.g. queries, classifications, etc.).

3.2.1.3 Exchange WMC file (Teamwork)

The tendency of specialisation in our society has fostered the development of domain experts in narrow niches. Therefore, it is necessary to cooperate within and across institutional borders. Information exchange and sharing is crucial. With the WMC (ViewContextCollection and ViewContext) this can be achieved. One possibility is to exchange a WMC "traditionally" as file. The other way is to share the information in almost "real time" between more than one person in a kind of "GIS conference". In both cases it is necessary to annotate, highlight and comment (Annotate / Highlight p. 46) specific features (anomalies) to discuss them either bilaterally or within an extended team.

3.2.1.3.1 Traditional information exchange

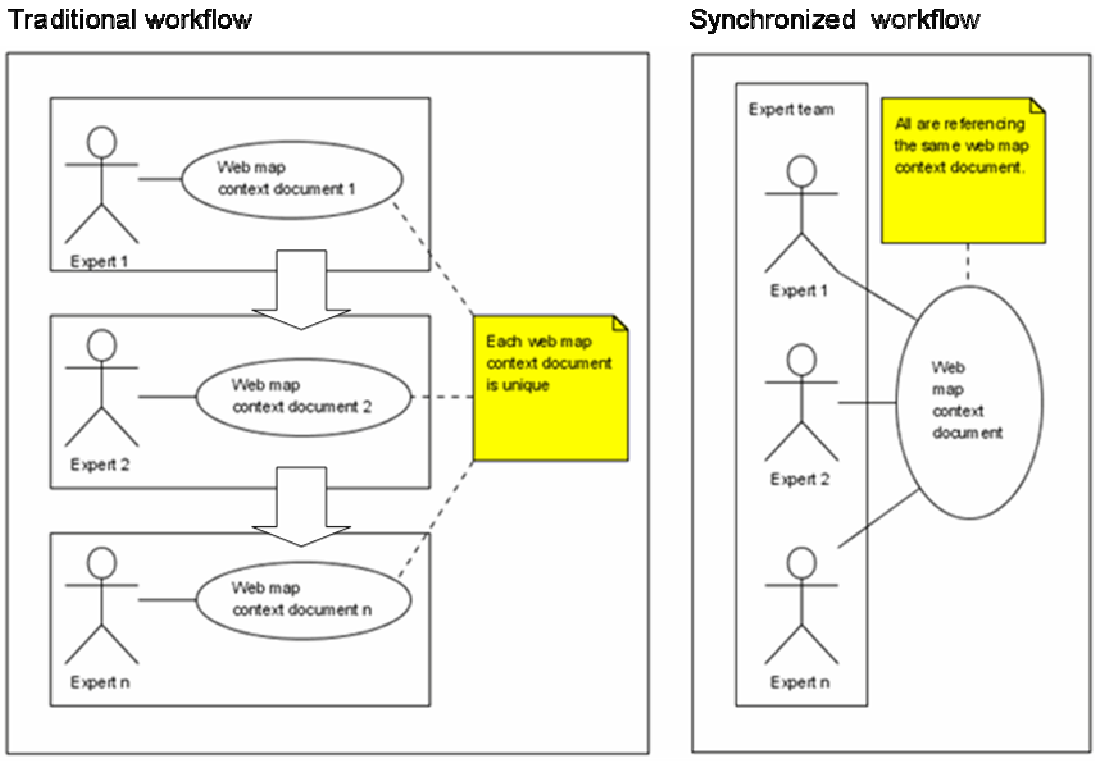


Figure 16 Synchronize ViewContext

As mentioned above, the traditional way is to exchange a WMC (ViewContextCollection, ViewContext) document as file. Once received the document is loaded into the WMC compliant software and processed further. Afterwards, the result is serialised (stored) and sent to the next expert and so on (see left side of Figure

16). Due to the fact that a “ViewContextCollection”, as well as a “ViewContext” document, are XML files they can be stored on any storage medium or be e-mailed as attachment. To do so it is necessary to be able to load (p. 31) and save (p. 39) a web map context document. The different experts involved comment and change the WMC document sequentially one after the other.

3.2.1.3.2 Synchronised – information sharing (“GIS conference”)

Traditionally, each user works with his own unique document (see left side of Figure 16). The innovation of “real time information sharing” is that the same WMC is shared simultaneously between multiple users (see right side of Figure 16). The difference in this case is that everybody uses the same “Web Map Context Document”. If anybody makes a change it is visible to all users that share the same WMC document.

3.2.2 Data storage, data maintenance, and data output

3.2.2.1 Serializing WMC files for later use (Store)

In order to reuse or share a WMC it is stored (serialised). This has to happen in a way that works independently of the file system and / or network protocols. This lies outside the specification but is essential for the daily work.

3.2.2.2 Edit and display (on output)

According to Tomlinson (2003) these functions are needed to create effective map products. A variety of editing, layout, symbolisation and plotting functions are therefore required.

3.2.2.2.1 Handling multiple views

Generally an expert needs multiple views during his work. An interactive map product usually contains multiple views. At least the detail map view, with which the user interacts and an overview map is needed (Figure 17, left side).

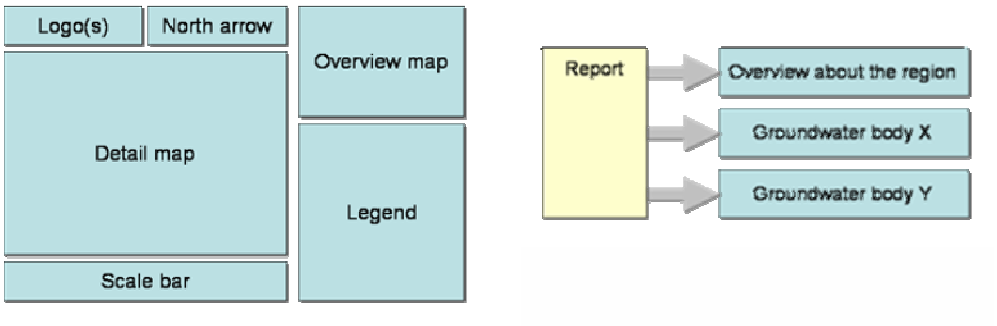


Figure 17 Multiple view examples: interactive map, report

Let's look in more detail at a report example (Figure 17, right side). The expert needs for his report three specific maps (view contexts), an overview of the region and two detailed maps with lists about the groundwater bodies of interest. Therefore, three view context documents have to be generated.

To store multiple ViewContexts the "ViewContextCollection" document is used. The elements of the ViewContextCollection document have already been discussed (see p. 31 for further information). Online referencing of views fosters the reuse and avoids duplication of work. Despite the advantage of reuse this approach introduces an additional problem. The respective ViewContext is referenced via "**U**niform **R**esource **L**ocator" (URL). This implicates it has to be accessible online. The access to detailed - in some cases sensitive – information, which may have to be restricted, causes additional security issues.

3.2.2.2.2 Visibility

Layers can be either displayed or hidden. The visibility change is reversible. The "hidden" property of the respective layer has to be set either to "true" or "false".

```
/ViewContext/LayerList/Layer[Name='The name of the layer']/@hidden
```

Figure 18 XPath expression: "hidden" attribute

The XPath (XML Path Language) is a query language for XML documents. It is used by the author to show the respective element and / or attribute.

3.2.2.2.2.1 Display layer

To display the layer the hidden property is set to "false".

```
<Layer queryable="true" hidden="false">
        … content of the layer …
</Layer>
```

Figure 19 Example "ViewContext" fragment: display layer

3.2.2.2.2.2 Hide layer

To hide the layer the hidden property is set to “true”.

```
<Layer queryable=”true” hidden="true">
        … content of the layer …
</Layer>
```

Figure 20 Example “ViewContext” fragment: hide layer

3.2.2.2.3 Add / Remove layer

With these functions the number of available layers in a “ViewContext” document is changed.

3.2.2.2.3.1 Add layer

```
<LayerList>                                   <LayerList>
        <Layer A>                                     <Layer A>
        <Layer B>          Add layer C →              <Layer B>
</LayerList>                                          <Layer C>
                                              </LayerList>
```

Figure 21 Example: Add layer

To add a layer means to insert a new record into the “LayerList” element. The therefore necessary meta-information is retrieved from the response of an OWS “GetCapabilities” request. The position in the “LayerList” where the “Layer” element is inserted determines the display position in the map (see “Change layer order” p. 42).

3.2.2.2.3.2 Remove layer

```
<LayerList>                                   <LayerList>
        <Layer A>                                     <Layer A>
        <Layer B>          Remove  layer B→           <Layer C>
        <Layer C>                             </LayerList>
</LayerList>
```

Figure 22 Example: Remove layer

This function simply deletes the “layer” element permanently from the “LayerList”. If the layer should only be hidden, the hidden attribute is used instead (see Hide layer p. 41).

3.2.2.2.4 Change layer order

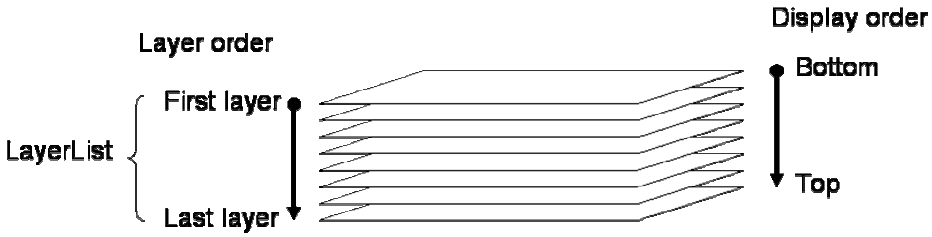


Figure 23 Layer visualizing order

The position of a "Layer" inside the "LayerList" element determines its display position. According to the "Web Map Context Documents" specification this follows a bottom to top approach (Humblet p. 14). The order of the layers can be simply changed by changing the position of a layer relative to the other ones.
E.g. If a layer is positioned after the last one (Figure 21 Example: Add layer*) it will be displayed on top position of the map.*

3.2.2.3 Symbolize

Tomlinson (2003) understands symbolisation as the process of selecting and using a variety of symbols to represent features as printed output as well as on the display.

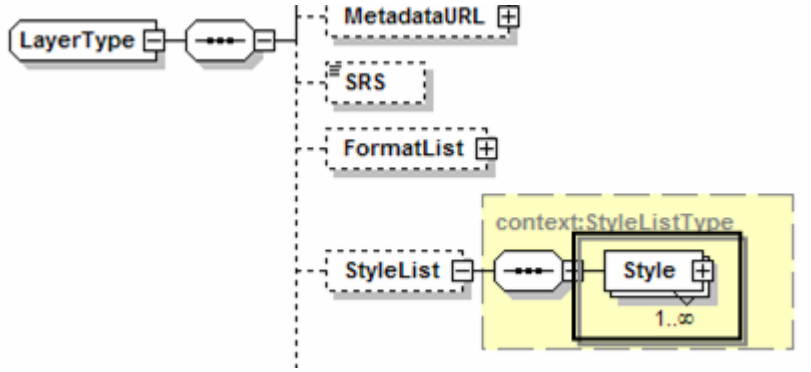


Figure 24 "ViewContext" XML schema fragment: "StyleList" and "Style" elements

Each layer can be visualised in many different ways. This is done with one or more "Style" elements.

3.2.2.3.1 Active style

```
/ViewContext/LayerList/Layer[Name='Layer name']/StyleList/Style[@current='true']
```

Figure 25 XPath expression: active style for "cite:BasicPolygons" layer

Only one style per layer can be active at once. The active style is determined by the "current" attribute. If the attribute is "true" the style is active.

3.2.2.3.2 *Predefined symbolisation*

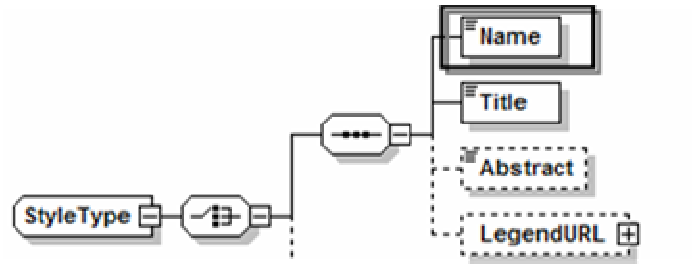


Figure 26 "ViewContext" XML schema fragment: Predefined symbolisation

A predefined symbolisation is a symbolisation (called "Style" in the WMS specification) which is defined on the WMS server. The user selects the style with the "Name". Every "Web Mapping Service" has to provide at least one default style.

```
/ViewContext/LayerList/Layer[Name='The layer name']/StyleList/Style[@current='true']/Name
```

Figure 27 XPath expression: predefined style of "cite:BasicPolygons" layer

```
<Style current="true">
   <Name>BasicPolygons</Name>
   <Title>Basic polygons</Title>
</Style>
```

Figure 28 Example "ViewContext" fragment: predefined style

3.2.2.3.3 *User defined symbolisation*

Sometimes it is necessary to define a new symbolisation style. This style can be either referenced online or be defined inline the "ViewContext" document. It can be symbolised with a "Styled Layer Descriptor" (SLD) enabled WMS server (Beaujardiere 2001 WMS, Lalonde 2002 SLD).

3.2.2.3.3.1 Online referenced resource

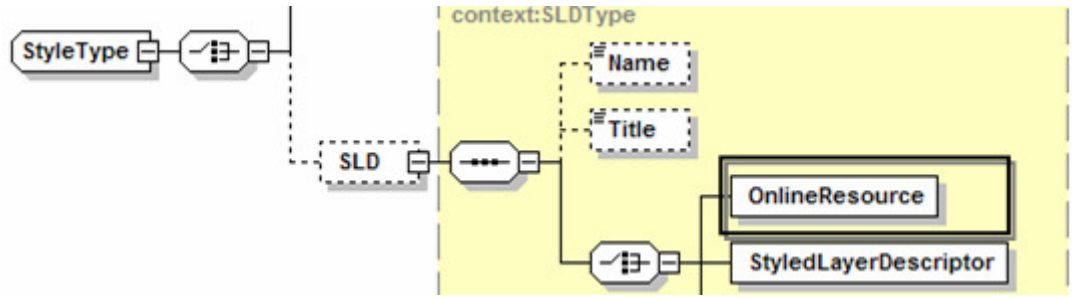


Figure 29 XML schema: User defined online resource symbolisation

```
/ViewContext/LayerList/Layer[/Name='Layer_name']/StyleList/Style[@current='true']/SLD/OnlineResource
```

Figure 30 XPath expression: User defined online resource symbolisation

```
<Style current="true">
        <SLD>
                <OnlineResource xlink:type="simple"
                xlink:href="http://127.0.0.1:8080/deegreewms/BasicPolygons.sld"/>
        </SLD>
</Style>
```

Figure 31 example "ViewContext" fragment: User defined online referenced SLD file

The referenced online resource, which is a "Styled layer descriptor file", can be reused from multiple users and for different purposes. It is possible to provide repositories of different styles for e.g. reporting obligations online accessible in the internet. If a style is changed the symbolisation is automatically updated for all users and purposes at once.

3.2.2.3.3.2 Inline definition

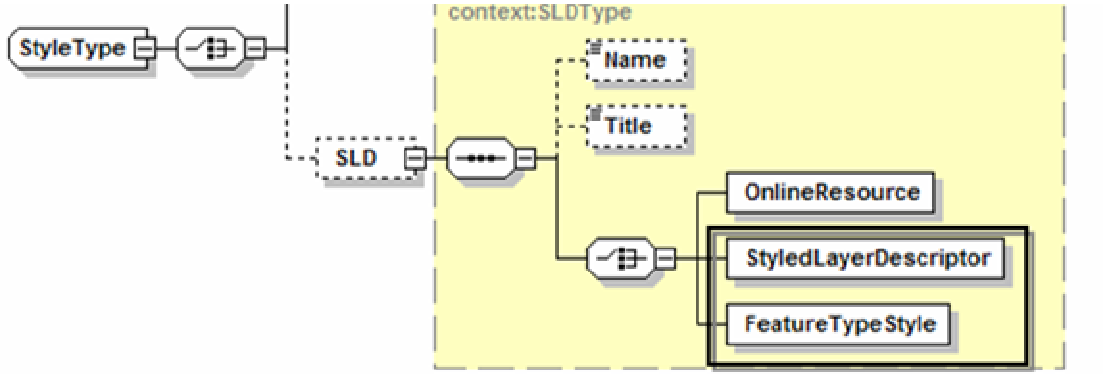


Figure 32 XML schema: User defined inline symbolisation

The inline declaration of a user-defined style allows a user specific style definition, which can be exchanged with the "ViewContext" document. This kind of style declaration is independent of online accessible SLD files. According to the SLD specification the user-defined style can be submitted to an SLD enabled WMS service with the help of the "SLD_BODY" element (et al Lalonde 2002, pp. 8-9). This can lead to troubles, if the size of the HTTP GET request exceeds the limit of the browser's buffer. The other possibility is to use an HTTP POST request where the size is principally unlimited. Unfortunately, the syntax of the POST request has not been specified within the WMS specification (Beajardiere 2002, p. 12) yet.

3.2.2.3.3.2.1 Inline defined "StyledLayerDescriptor"

In this case the StyledLayerDescriptor is declared inline in the ViewContext document.

```
/ViewContext/LayerList/Layer[/Name='Layer_name']/StyleList/Style[@current='true']/SLD/StyledLayerDescriptor
```

Figure 33 XPath expression: User defined inline symbolisation with "SLD"

```
<Style current="true">
        <SLD>
                <sld:StyledLayerDescriptor version="1.0.0" >
                        … the content …
                </sld:StyledLayerDescriptor>
        </SLD>
</Style>
```

Figure 34 Example "ViewContext" fragment: User defined inline "SLD"

3.2.2.3.3.2.1.1 Remark

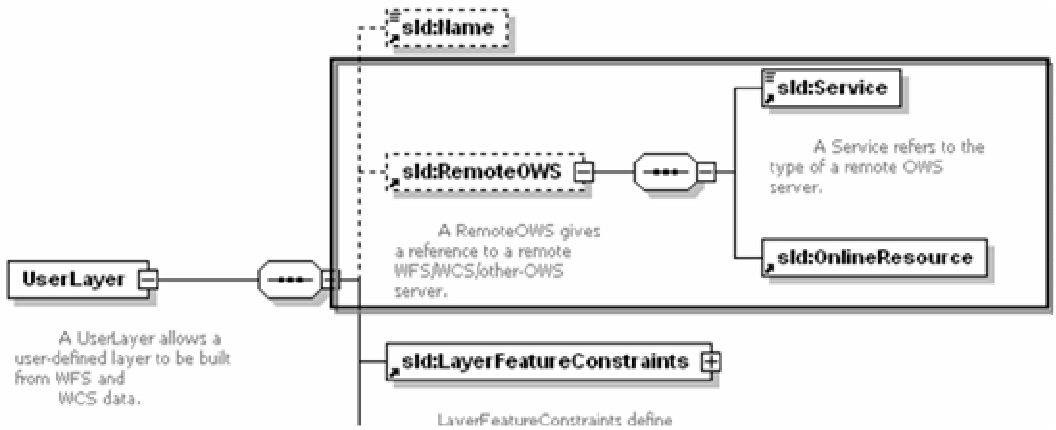


Figure 35 "SLD" XML schema fragement: Remote OWS server

With the inline as well as with the online referenced "StyledLayerDescriptor" it is possible to use a remote OWS (OGC Web service) enabled "Web Mapping Service" (integrated and component server) server to visualize a WFS or WCS service in a map. This is crucial, because it allows to add a WFS / WCS service (component server) on-the-fly. Furthermore, a WFS service allows the generation of lists (p. 55) and graphs (p. 59).

3.2.2.3.3.2.2 Inlinde defined "FeatureTypeStyle"

The "FeatureTypeStyle" is a fragment of a correct SLD file declared inline a ViewContext document.

```
/ViewContext/LayerList/Layer[/Name='Layer_name']/StyleList/Style[@current='true']/SLD/FeatureTypeStyle
```

Figure 36 XPath expression: User defined inline symbolisation with "FeatureTypeStyle"

```
<Style current="true">
  <SLD>
    <sld:FeatureTypeStyle>
        … the content …
    </sld:FeatureTypeStyle>
  </SLD>
</Style>
```

Figure 37 "ViewContext" document fragment: User defined inline FeatureTypeStyle

3.2.2.3.4 Annotate / Highlight

"Imagery analysts, and other users of imagery and maps, often need to summarize the essential content of an image, point out features of interest, or express similarities or differences between images. Likewise, GIS specialists often need to highlight spatial patterns, label certain features, or otherwise "mark up" a map. Any of these activities creates and communicates a valuable piece of information: an image or map annotation." (Evans 2001, p. 2)

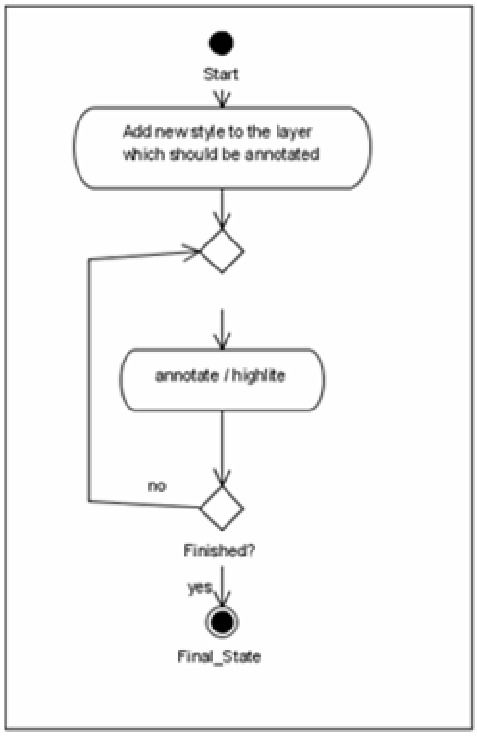


Figure 38 Annotation workflow

An empty layer is added. It references the layer which should be annotated. This new layer contains the annotation entries which are defined inline in its "StyledLayerDescriptor". Further annotation may be added interactively. The whole functionality of a SLD file can be used, which makes it possible to highlight features, add comments, sketches and last but not least graphs. The new annotated layer is displayed above the layer it references.

3.2.2.4 Plot

Sophisticated maps displayed only on a computer screen are useless if you cannot output them on paper. This is usually done in a user defined scale, paper format and output quality. Therefore, it is necessary to define how large a pixel on the output device will be (printer or plotter).

3.2.2.4.1 Plot a given extent in a specified output quality and scale

The common workflow is to define the scale which determines the size of the hardcopy output map on the paper. The required map size in pixel is derived by multiplying the output map width and height with the output quality. In the last step the image is displayed with the help of a "Cascading Style Sheet" (CSS) which allows specifying the exact width and height.

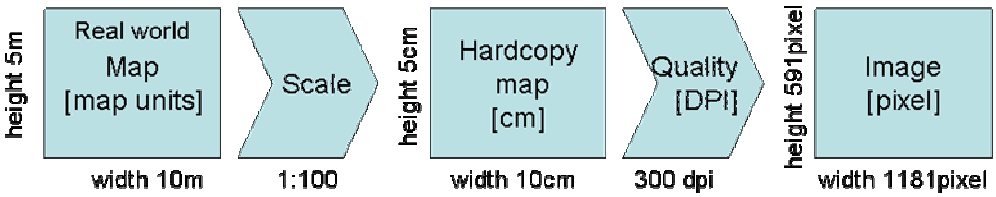


Figure 39 Steps to plot a given extent (bounding box) in a specific scale

Real distance	Map distance	Map distance	Output	Map width / height
dist	$mapDist = \frac{dist}{100}$	$inchMapDist = \frac{mapDist}{0.0254}$	*dpi*	*inchMapDist * dpi*
[m]	[m]	[inch]	[dpi]	[pixel]
10	0.10	3.94	300	1181
5	0.05	1.97	300	591

Figure 40 Calculation required for producing scaled map on paper

```
<img src="The getMap request" style="width:10cm;height:5cm;"/>
```

Figure 41 "img" element on website

Geographic coordinate systems (angular unit degrees) have to be converted into projected coordinate system (linear unit meters) before this process can take place.

3.2.2.4.2 Plot a given map width and height on the paper in a specified output quality and scale

In this case the extent is scaled according to the specified width and height. The centre remains unchanged. The output quality once more determines the needed pixel width and height of the output map. In this case the output width and height are fixed and the extent (bounding box) is changed accordingly.

3.2.2.5 Browse

3.2.2.5.1 Convert

User interaction happens with the screen e.g. the user clicks with the mouse on a pixel on the screen and wants to identify a feature. Therefore, it is necessary to be able to convert from screen units (pixel) into map units.

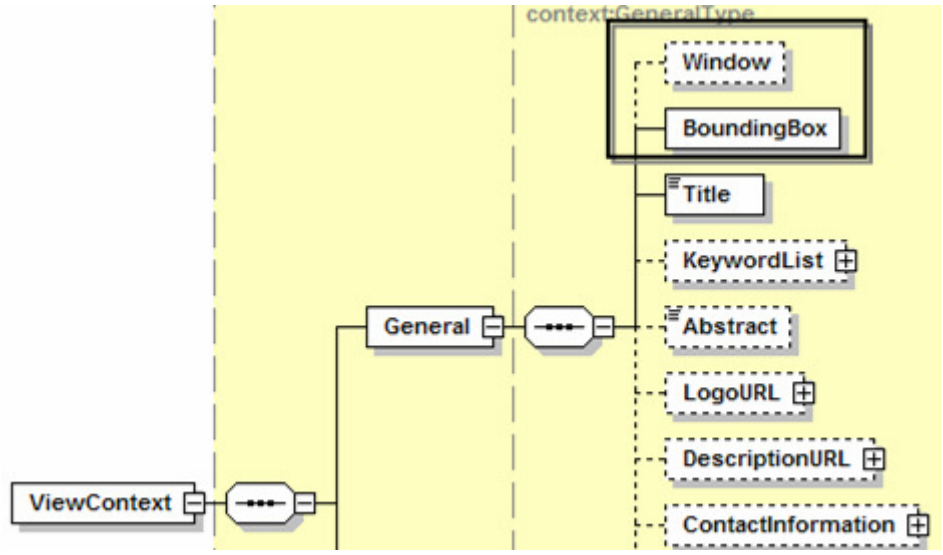


Figure 42 "ViewContext" XML schema fragment: "BoundingBox" and "Window" element

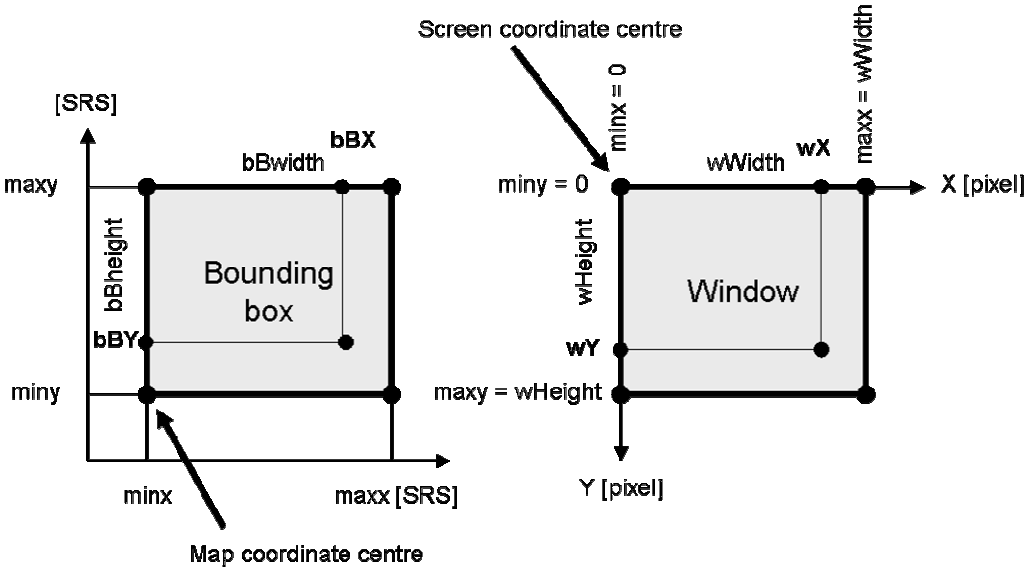


Figure 43 BoundingBox versus Window element

Note by the authors: It has to be kept in mind, that the coordinate centre is different between the screen and the map (Figure 43).

/ViewContext/General/Window

Figure 44 XPath expression: "Window" element

The "Window" element contains the width and height in pixel ("width" and "height" attribute) of the map displayed on the screen.

/ViewContext/General/BoundingBox

Figure 45 XPath expression: "BoundingBox" element

The “BoundingBox” element contains the extent (minx, miny, maxx and maxx attributes) of the displayed bounding box and the spatial reference system (SRS attribute) which determines the map units.

3.2.2.5.1.1 Convert pixel distance into to map units

This function is required for moving the bounding box dx pixel in x and dy pixel in y direction.

$bDx = \frac{dx \times wWidth}{bWidth}$	The distance in map units in x direction.
$bDy = \frac{dy \times wHeight}{bHeight}$	The distance in map units in y direction.

Figure 46 Formula: convert dx, dy from pixel into mapunits

3.2.2.5.1.2 Convert pixel coordinates into map units

This function converts the position of a point on the screen from pixel into map units. It is required to measure distances on the screen.

$bBX = \min x + wX \times \frac{bWidth}{wWidth}$	The x coordinate in map units.
$bBY = \max y - wY \times \frac{bBHeight}{wHeight}$	The y coordinate in map units.

Figure 47 Formula: convert point from pixel into map units

3.2.2.5.1.3 Convert BoundingBox to map units

The same formula as above is used for getting minx, miny, maxx, and maxy values. This function is required for zooming in with a given rectangle.

3.2.2.5.2 Move

The move functions allow to change the displayed map extent (bounding box) of a “ViewContext”. The width and height of the bounding box remains unchanged. All “move” use cases are based on the same function, which adds a dx value to the minx and maxx attribute and / or a dy value to miny and maxy attribute of the “BoundingBox” element. In case dx and dy are given in pixel (screen units) their values have to be converted into map units before the move process starts.

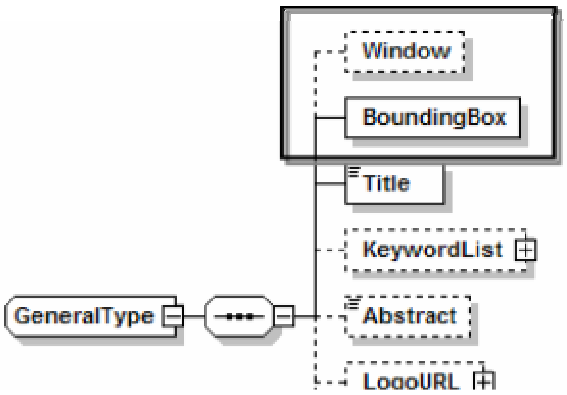


Figure 48 "ViewContext" XML schema fragment: required elements for move function

/ViewContext/General/BoundingBox

Figure 49 XPath expression: "BoundingBox" element

3.2.2.5.2.1 Move dx, dy

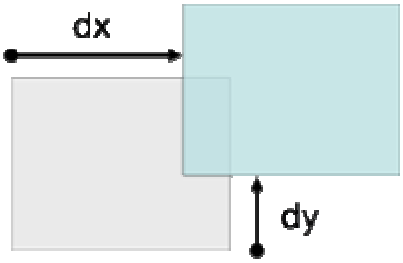


Figure 50 Move dx, dy

Moves the visible map extent dx map units westwards and dy map units northwards.

minx = minx + dx

maxx = maxx + dx

miny = miny + dy

maxy = maxy + dy

3.2.2.5.2.2 Move from point to point (pan)

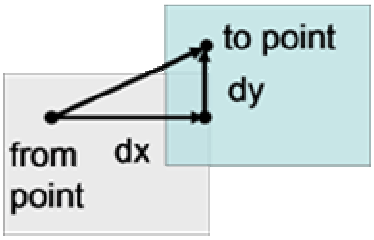


Figure 51 Move from point to point (pan)

Moves the visible map extent along a line from a given starting point to a given endpoint.

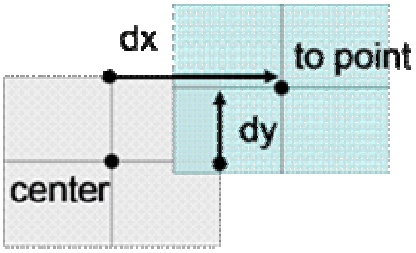
dx = toPointX – fromPointX

dy = toPointY – fromPointY

→ Move dx, dy

3.2.2.5.2.3 Move to point (centre at)

This function recentres the displayed extent. It is only a special case of moving from a starting towards an endpoint. In this case the starting point is the centre of the present extent.



$$fromPo\,\text{int}\,X = \min x + \frac{\max x - \min x}{2}$$

$$fromPo\,\text{int}\,Y = \min y + \frac{\max y - \min y}{2}$$

→ Move from point to point

Figure 52 Move to point (centre at)

3.2.2.5.3 Identify

Identify is the function which provides interaction with the map content. Therefore, all the required input parameters of a WMS "GetMap" request (Beaujardiere 2002, Table 9 p. 40) are necessary. Additionally, the position (x and y coordinate in pixel) where the information is requested has to be provided (Beaujardiere, 2002).

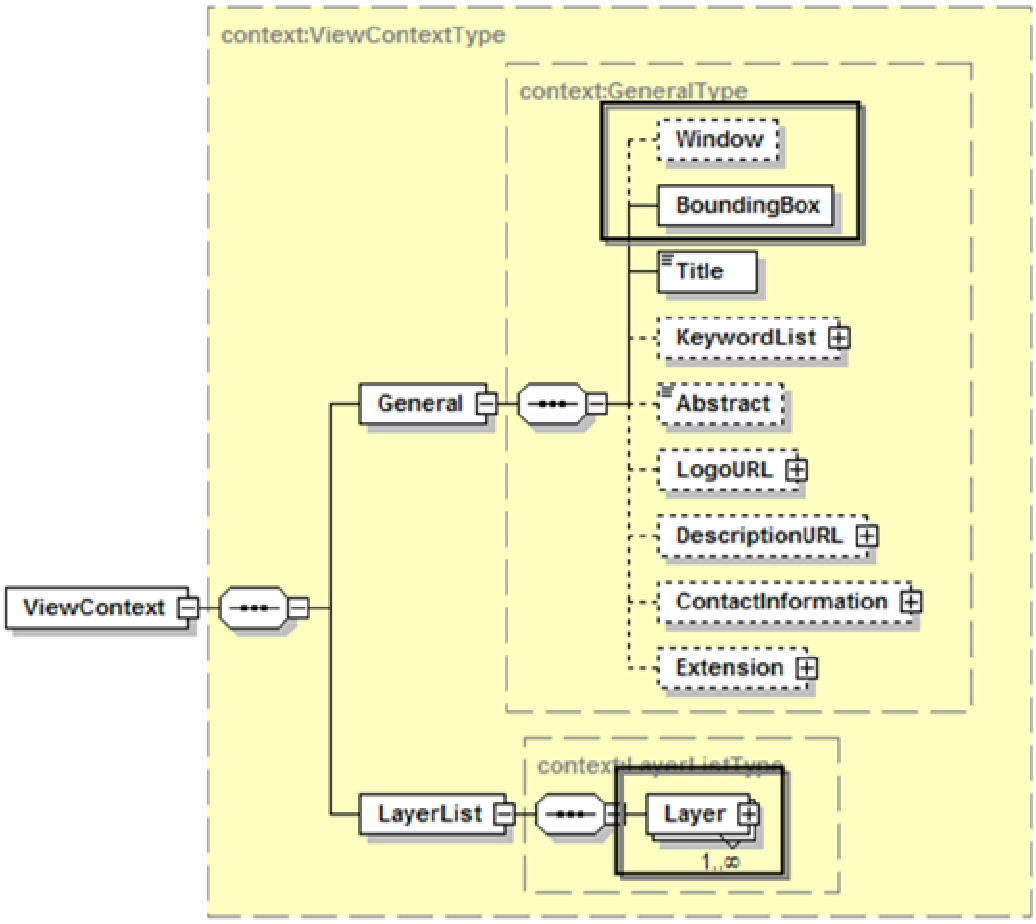


Figure 53 "ViewContext" XML schema: Required elements for map interaction

The "Window" element contains the width and height of the map on the screen in pixel, the "BoundingBox" the map extent and spatial reference system and the "Layer" the necessary information to access and use the layer (service URI, format, style).

The layers which are queryable (interactable) are marked with the “queryable” attribute which is set to “true”.

```
<Layer queryable=”true” hidden=”false”>
        … the layer content …
</Layer>
```

Figure 54 ”ViewContext” document fragment: queryable attribute

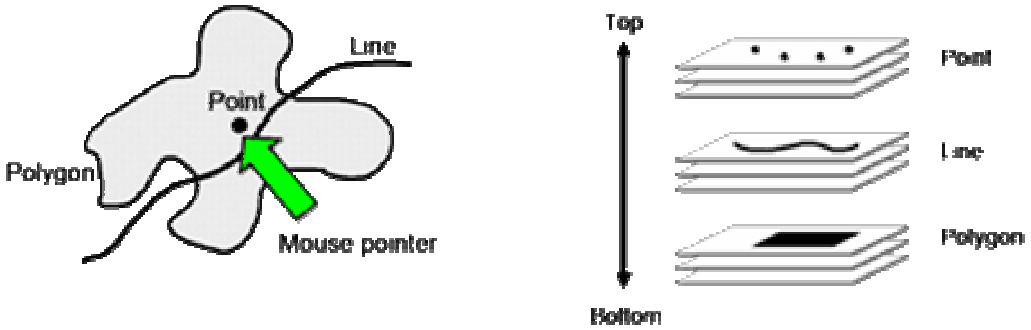


Figure 55 Identify a feature

To identify a feature in the map, the user positions the mouse pointer on the feature and clicks on the mouse button. It is the task of the software to determine the features whose attribute information will be shown. The best way to do so is to reorder the queryable and visible Layers. On top are the points, followed by the polyline and finally at the bottom by the polygon layers. Unfortunately, due to the fact the “ViewContext” document is not able to store the geometric feature type in its meta-information properties (p. 35) this approach is not possible. The only alternative is to query all layers with the “queryable” attribute set to true. The technical requirement to perform the identify operation is a “GetFeatureInfo” enabled WMS web server.

```
/ViewContext/LayerList/Layer[@queryable='true']
```

Figure 56 XPath expression: Get all query able layers

3.2.2.6 Suppress

According to Tomlinson (2003) this function removes features from the working space. They are omitted in the subsequent manipulation and analysis processes. The “Styled Layer Descriptor” (SLD) specification allows with the “LayerFeatureConstraints” element to define which features are retrieved from a layer. This is also important to minimize the amount of data that has to be transferred e.g. from a remote WFS server to a WFS enabled WMS server. The technical prerequisite for suppress is a SLD enabled “Integrated” or “Component” WMS server (p. 30).

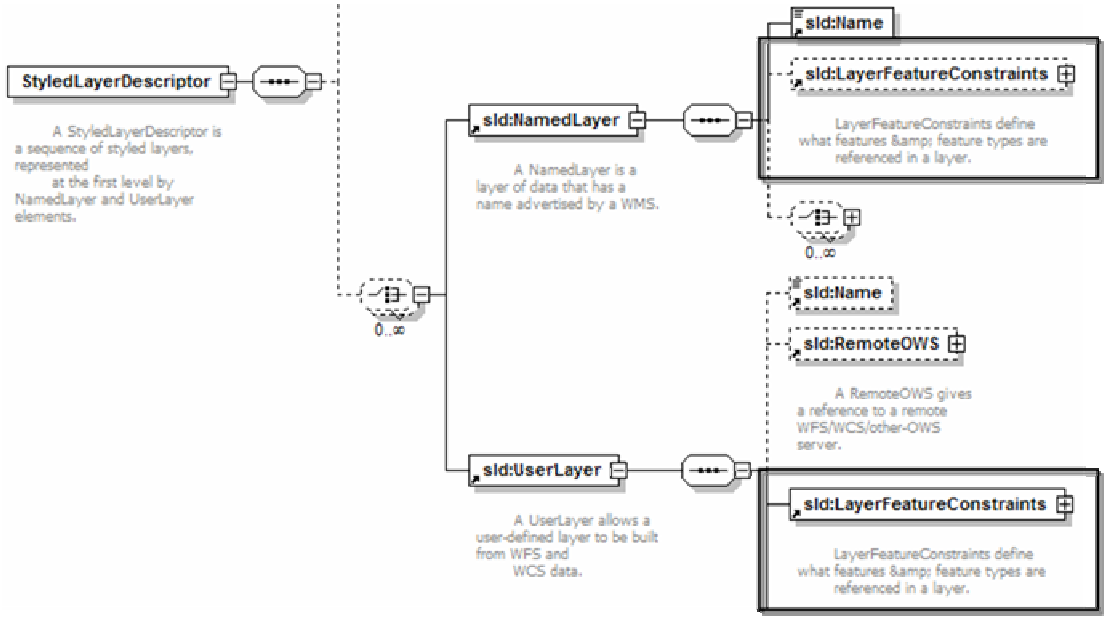


Figure 57 ”SLD” XML schema fragment: “LayerfeatureConstraints” element

/ViewContext/LayerList/Layer[/Name=’The_layer_name’]/StyleList/Style[/Name=’ The style name’]/SLD/sld:StyledLayerDescriptor/sld:NamedLayer/sld:FeatureTypeContraints

Figure 58 Xpath expression: “FeatureTypeContraints” in “NamedLayer”

/ViewContext/LayerList/Layer[/Name=’The_layer_name’]/StyleList/Style/SLD/sld:StyledLayerDescriptor/sld:UserLayer/sld:FeatureTypeContraints

Figure 59 XPath expression: “FeatureTypeContraints” in “UserLayer”

The “FeatureTypeContraints” element allows suppressing (which is basically a query) for a predefined “NamedLayer” provided by a WMS service as well as for a user defined “UserLayer”.

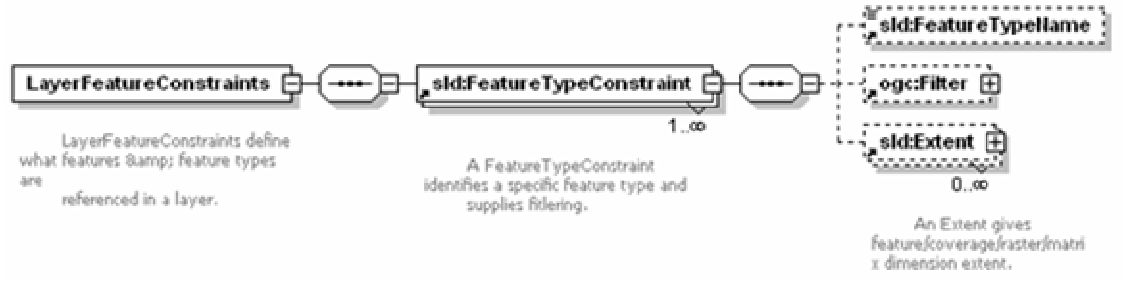


Figure 60 ”StyledLayerDescriptor” XML schema: LayerFeatureConstraints element

The element allows to restrict the features by a given filter and / or extent. This cannot only be used for suppressing features, furthermore, it can be used to store user defined queries. See “Query” chapter (pp. 56) for further details. All additional queries are added with the “ogc:And” filter.

3.2.2.7 Create list (report)

This function creates information products (IP) in the form of lists and / or reports. The current WMC specification is limited to WMS services. In order to create such an IP it is additionally necessary to access the layer as OWS service (e.g. WFS or WCS).

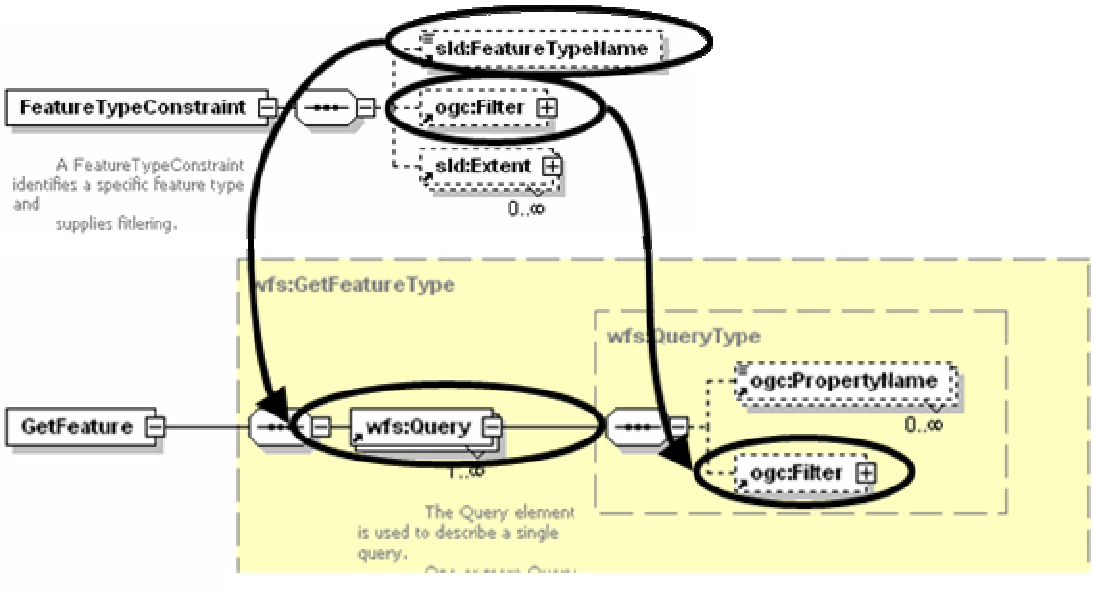


Figure 61 "SLD" / "WFS" XML schema fragement: Construct WFS request

Figure 62 Steps to generate a list / report

The target server and service type (either WFS or WCS) can be derived from the RemoteOWS element (see page 35). The "GetFeature" request is composed of the "FeatureTypeName" and the given "ogc:Filter" (Query pp. 56). The response is a XML file in the "Geographic Markup Language" (GML) format. As every XML file, it can be easily processed and further transformed with an "Extensible Stylesheet Language" (XSL) style sheet into the format of the user's choice.

```
<LayerFeatureConstraints>
  <FeatureTypeConstraint>
    <FeatureTypeName>topp:states</FeatureTypeName>
    <ogc:Filter>
      <ogc:PropertyIsLike wildCard="*">
        <ogc:PropertyName>topp:STATE_NAME</ogc:PropertyName>
        <ogc:Literal>T*</ogc:Literal>
      </ogc:PropertyIsLike>
    </ogc:Filter>
```

```
    </FeatureTypeConstraint>
</LayerFeatureConstraints>
```

Figure 63 Example:"LayerFeatureConstraints"

3.3 Data manipulation and analysis functions

3.3.1 Query

According to Tomlinson (2003) querying is the process of selecting a subset of a dataset based on questions about their spatial and / or attributive characteristics. The resulting subset can be used for reporting, further study, or analysis.

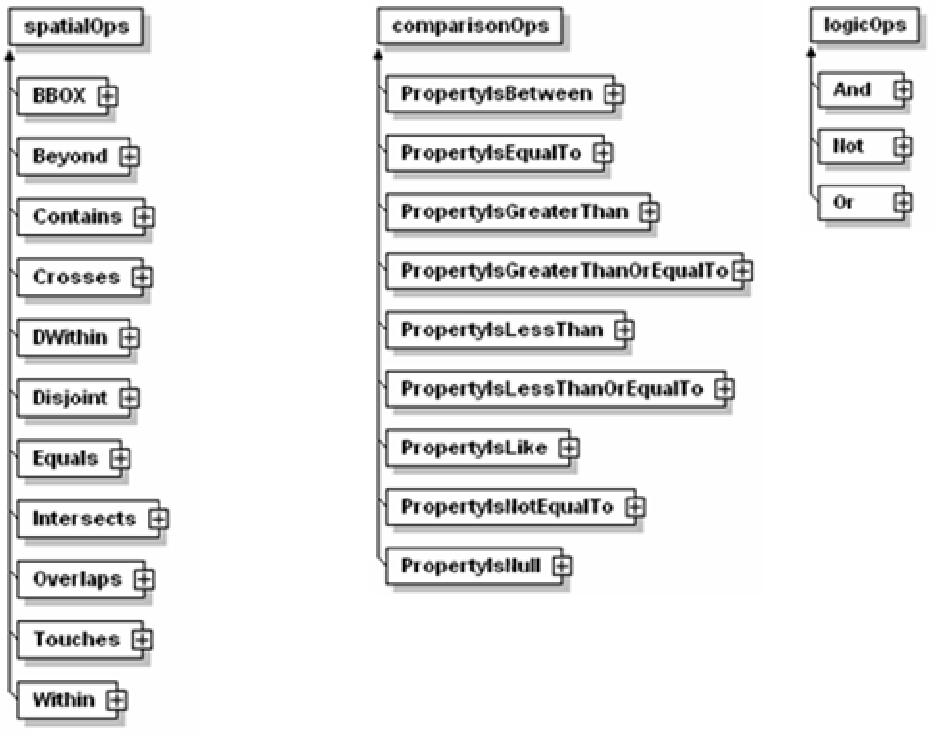


Figure 64 Available filter operators according OGC Filter encoding specification

The Figure 64 above shows filtering operators defined in the OGC "Filter" specification (Vretanos 2001, pp. 9-19). The filters are divided into spatial, comparison and logical operators.

Figure 65 "StyledLayerDescriuptor" XML schema: query place

The query is inserted under the “LayerFeatureTypeConstraints” element. This works only with a SLD enabled “integrated” and or “component” WMS server (p. 30). The result can be displayed as map as well as list / report if the layer is available as OWS service.

3.3.1.1 Spatial query (Spatial operators)

Spatial query is the process of selecting a subset of a dataset with the help of spatial criteria. The most common queries are possible with the available spatial operators (Figure 64). They are described in greater detail in the “Simple feature for SQL specification Version 1.1” (Beddoe 1999, pp. 2).

```
<ogc:Filter>
  <ogc:BBOX>
    <ogc:PropertyName>/Europe/Border</ogc:PropertyName>
    <gml:Box>
      … The geometry …
    </gml:Box>
  </ogc:BBOX>
</ogc:Filter>
```

Figure 66 Example: spatial query

In this case all features are selected, that are within the given bounding box.

3.3.1.2 Attribute query (Comparison operators)

An attribute query filters by comparing attribute values of features. The most common queries are possible with the in the filter specification available comparison operators (Figure 64).

```
<ogc:Filter>
  <ogc:PropertyIsBetween>
    <ogc:PropertyName>The property name</ogc:PropertyName>
    <ogc:LowerBoundary>
      <ogc:Literal>The value</ogc:Literal>
    </ogc:LowerBoundary>
    <ogc:UpperBoundary>
      <ogc:Literal> The value </ogc:Literal>
    </ogc:UpperBoundary>
  </ogc:PropertyIsBetween>
</ogc:Filter>
```

Figure 67 Example: attribute query

3.3.1.3 Combination (Logical operators)

The possibility to combine the query type with logical operators (“And”, “Or”, “Not”) in a user defined way offers very advanced query capabilities.

```
<ogc:Filter>
        <ogc:And>
                Condition A
                <ogc:Or>
                        Condition B
                        Condition C
                </ogc:Or>
        </ogc:And>
</ogc:Filter>
```

Figure 68 Example: combined query

In this example A and either B or C must be true.

3.3.2 Generating features, views, and graphs

3.3.2.1 Generate buffer

According to Tomlinson (2003), the buffer function allows generating zones around points, lines and / or polygon features It allows identifying features that are within (DWithin) or outside (Beyond) such a zone.

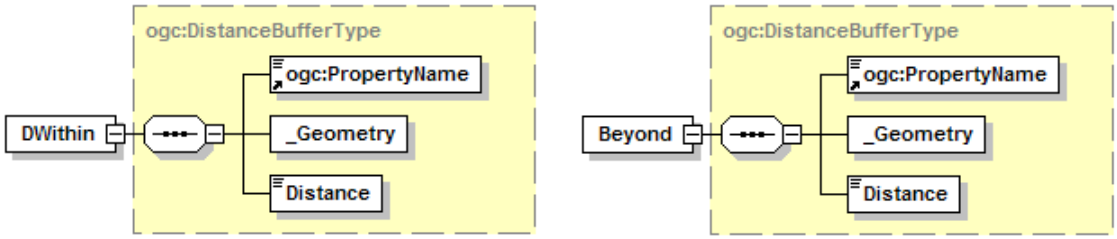


Figure 69 DWithin and Beyond element

```
<ogc:Filter>
  <ogc:DWithin>
    <ogc:PropertyName>The property name</ogc:PropertyName>
    <gml:Point>
      … The geometry …
    </gml:Point>
    <ogc:Distance units="http://www.umodict.com/uom.html#meters">The value</ogc:Distance>
  </ogc:DWithin>
</ogc:Filter>
```

Figure 70 Example: buffer

3.3.2.2 Generate graph

This function provides the ability to generate a graph from attribute data. A graph may display two or more attributes (e.g. pie chart, column chart, network chart, etc.).

Figure 71 Steps to generate a graph

This capability is not covered by the specification. But it is possible to generate the graphs from the list output in spreadsheet software and / or develop a special component. The prerequisite is once more that the layer is either available as WFS for vector datasets, or WCS service for raster datasets.

3.3.3 Manipulating features

3.3.3.1 Classify attributes

It is very easy to classify (group) similar features according to their attributes into user defined classes with the help of a "Styled Layer Descriptor" (SLD).

```
<Rule>
   <ogc:Filter>
         … Filter criteria …
   </ogc:Filter>
   <PointSymbolizer>
         … The symbolizer …
    </PointSymbolizer>
</Rule>
```

Figure 72 Example: classify attributes with SLD file

For each defined class a rule is created. Each rule contains filter criteria (usually comparing operators e.g. "ogc:PropertyIsBetween"). All features that comply with this filter are visualised according to the given symboliser. All operators explained in the query chapter (p. 56) can be used here. The whole SLD capabilities can be applied. This allows to use even rather complex compound symbolisers. It is important to keep in mind, that if an operator is valid in multiple rules, the feature is symbolised for all matching rules. In these cases the technical prerequisite is a SLD enabled WMS server that allows user defined symbolisation.

3.3.3.2 Clip

According to Tomlinson (2003) clipping allows to extract features from a defined area from the data source.

E.g. extract all sampling sites within the county XY boundary.

The mechanism is the same as with query (p. 56) and suppress (p. 53). In this case the respective filter is the geometry of the county XY. If a filter has already been defined in the "LayerFeatureConstraints" element, the spatial operator is added with the logical "ogc:And" operator. Once more a SLD enabled WMS is required.

3.3.3.3 Scale change

Changing the scale means to change the size in which the features are displayed. This action can only be done on the computer. An output like a hardcopy has only one scale which cannot be changed (Tomlinson, 2003). If the centre or the bounding box - to which should be scaled - is given in pixel their values have to be converted into map units before the scaling process can begin.

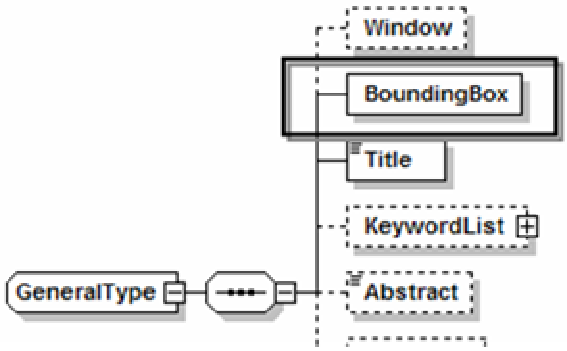


Figure 73 "ViewContext" XML schema: required input for scale change

/ViewContext/General/BoundingBox

Figure 74 XPath expression: BoundingBox element

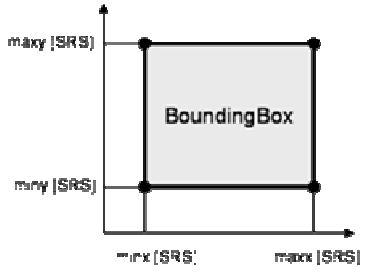


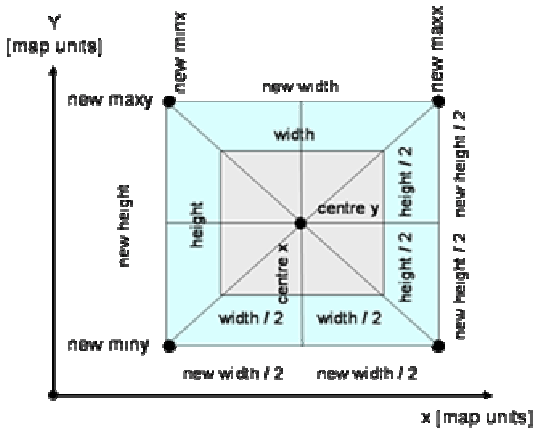
Figure 75 BoundingBox element

Scaling is the process that changes the extent (bounding box) of the displayed view context.

The output width and height on the display or on paper remains unchanged.

3.3.3.3.1 Scale width, height

This function leaves the centre unchanged. The width and height of the bounding box is scaled and therefore its minx, miny, maxx and maxy attributes are changed.



$$new\min x = \frac{centre\,x - new\,width}{2}$$

$$new\min y = \frac{centre\,y - new\,height}{2}$$

$$new\max x = \frac{centre\,x + new\,width}{2}$$

$$new\max y = \frac{centre\,y + new\,height}{2}$$

Figure 76 Scale width, height

3.3.3.3.2 Scale factor (zoom in and zoom out)

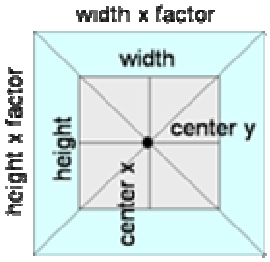


Figure 77 Scale to factor (zoom in and zoom out)

Once more the centre remains unchanged and the width and height is multiplied with the given factor. If the factor is below one the user zooms in. If it is greater than one the user zooms out.

$$new \quad width = factor \times width$$

$$new \quad height = factor \times height$$

→ The next step is to scale width, height

3.3.3.3.3 Scale centre, factor

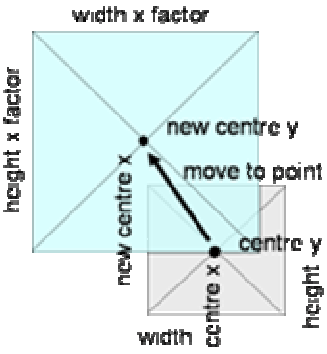


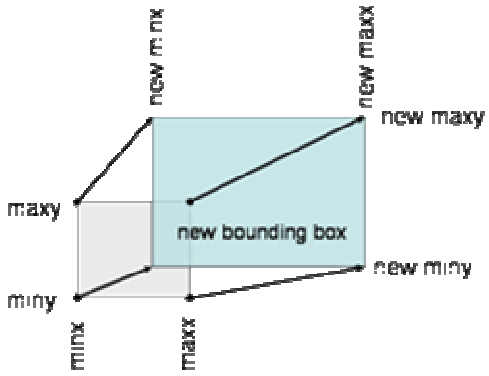
Figure 78 scale centre, factor

In this case the new centre and a factor are defined. It is a combined function.

→ First the bounding box is moved to the new centre ("move to point" function).

→ Afterwards the "scale factor" function is invoked.

3.3.3.3.4 Scale bounding box



Each point of the bounding box is given a new value. This function is invoked if the user scales to a user defined rectangle.

minx = new minx
miny = new minx
maxx = new maxx
maxy = new maxy

Figure 79 Scale to bounding box

3.3.3.3.5 Scale to exact scale

According to the SLD specification (Lalonde p. 27) the pixel size of the video display is unknown. Therefore, it is assumed that the "standard pixel rendering" size is 0.28mm. It is not possible to provide an exact scale on an unknown display unit. In principle the same approach as for plotting in a given scale could be applied here (p. 47).

3.3.3.4 Projection change

If the projection is changed the spatial reference system property (SRS) of the "BoundingBox" element is set to the new value.

/ViewContext/General/BoundingBox/@SRS

Figure 80 XPath expression: "SRS" element

The spatial reference system can be changed if all OGC OWS which are used in this view context, support this system.

/ViewContext/LayerList/Layer[/SRS='the SRS attribute of the BoundingBox element']/Name

Figure 81 XPath expression: layer that support the given SRS

The XPath expression returns all layers of the "LayerList" that support the respective SRS system. Unfortunately, the specification of the XML schema has an error. It allows only to store one spatial reference system (p. 35).

3.3.4 Address locations

3.3.4.1 Address geocode (search)

According to Tomlinson (2003), the address geocode function allows adding point locations derived from address information to a map. The author has extended this definition with a search for features capability. The prerequisite to do this is a SLD enabled "Internal" or "Component" WMS server that exposes the layer additionally as WFS service.

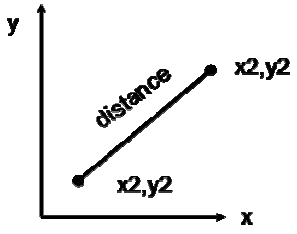
Figure 82 Address geocode workflow with a WFS service

The first step is to query the WFS service for the feature. All filter capabilities can be applied (attributive, spatial and logical operators). The response contains the bounding box of the geometry. If it is a point that has no extent, a generic approach is needed. A buffer has to be added around the feature (e.g. 10km). Additionally, it is necessary to visually identify the feature. It has to be highlighted with a "Styled Layer Descriptor" file (Annotate / Highlight p. 46). The WMS "GetMap" request with the "buffered" bounding box and the user defined "Styled Layer Descriptor" is performed. The result is a map zoomed in towards the highlighted geocoded feature.

3.3.5 Measurement

3.3.5.1 Measure length

This function allows to measure the length of a line. This is usually done by interactive user input on the screen. The input is retrieved in pixels and has to be converted into map units (See "Convert" chapter p. 48 for more details).



With converted point coordinates the distance can easily be calculated. It has to be kept in mind, that projections are not always equidistant. So in these cases it can be only a rough estimation.

$$distance = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$$

Figure 83 Measure distance

In the case of a polyline this approach is repeated for each line segement.

3.3.6 Spatial Analysis

3.3.6.1 Graphic overplot

Graphic overplot superimposes one layer on another one. It shows the intersection between the datasets. It is commonly used to get a visual impression about the topological relationship of datasets.

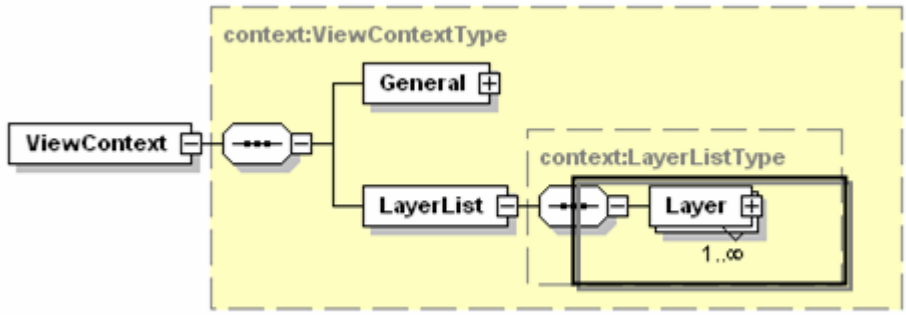


Figure 84 ViewContext XML schema: Layer element

Each layer is separately requested as image from the WMS server. Beginning with the first layer (= bottom) (p. 42) all layers are visualised according to their current active styles and formats. It is important that the layer's background is transparent, because otherwise it is not possible to overlay them.

4 Implementation

After the theoretical review the next step of the author's approach is to implement the as required identified functionality. The OGC usually specifies its services in a written documentation of the functionality and with "Extensible Markup Language" (XML) schemas. Depending on the topic the specifications may change frequently. In most cases one specification references and reuses other specifications. This applies for the "Web Map Context Documents" specification, too. The implementation starts with a review of the WMC XML schema and continues with an overview of the author's implementation approach. The technology used is only shown in principle, because it goes beyond the scope of the thesis, which aims at evaluating the specification and is not primarily a software development project.

4.1 Review of WMC XML schema

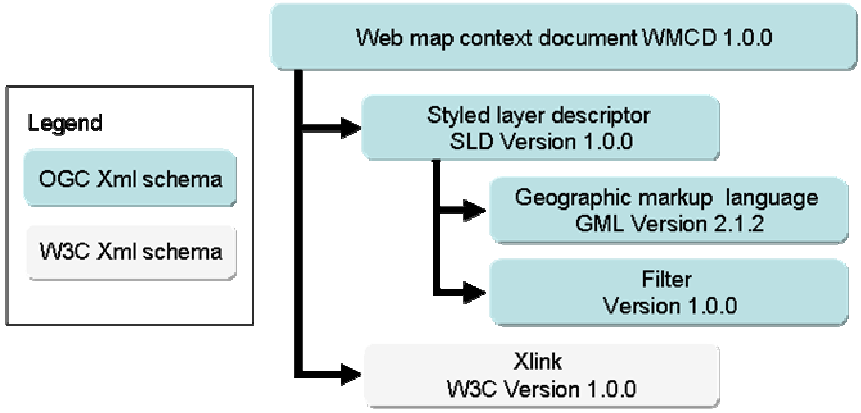


Figure 85 "Web Map Context Documents" referenced XML schemas

The WMC XML schema references the "Styled Layer Descriptor" (SLD) schema, which indirectly references the "Geographic Markup Language" (GML) and the "Filter" schema. Additionally, the "World Wide Web Consortium" (W3C) "Xlink" schema is also referenced.

4.2 Implementation overview

Due to the fact that specifications change often the author used a flexible mechanism to map a XML schema into the Java programming language.

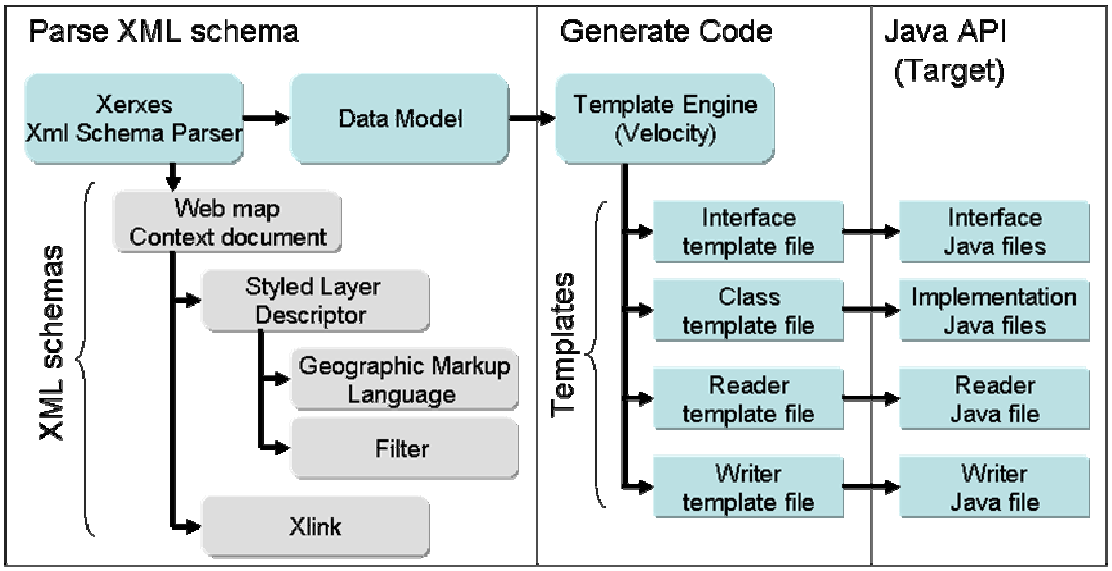


Figure 86 XML schema mapping workflow

The implementation parses the "Web Map Context Documents" XML schema ("context.xsd" and "collection.xsd") and all of its referenced XML schemas. The result is a data model.

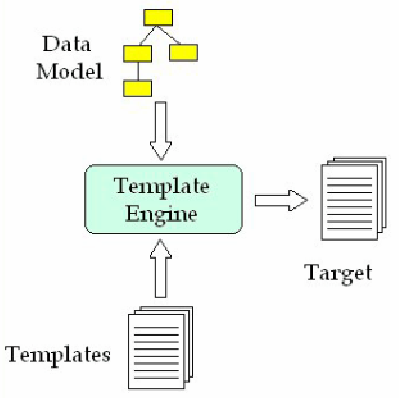
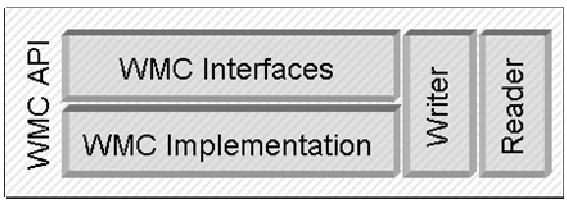


Figure 87 Template engine (Naccarato 2004)

This data model is mapped with the help of the "Velocity" template engine into a target (Figure 87).



The resulting "target" is a "Web Map Context Documents" application programming interface (API) that allows reading, writing and constructing "ViewContext" and

Figure 88 WMC API (Target)

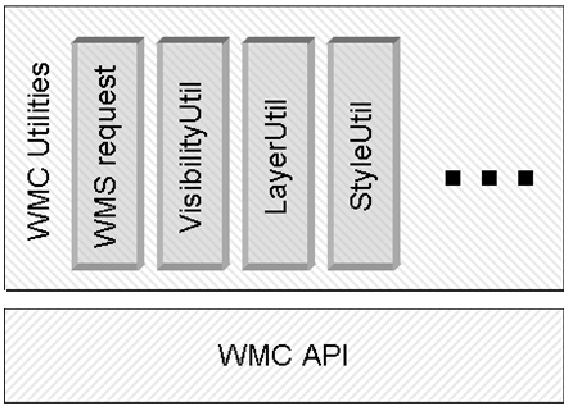


Figure 89 Utility classes

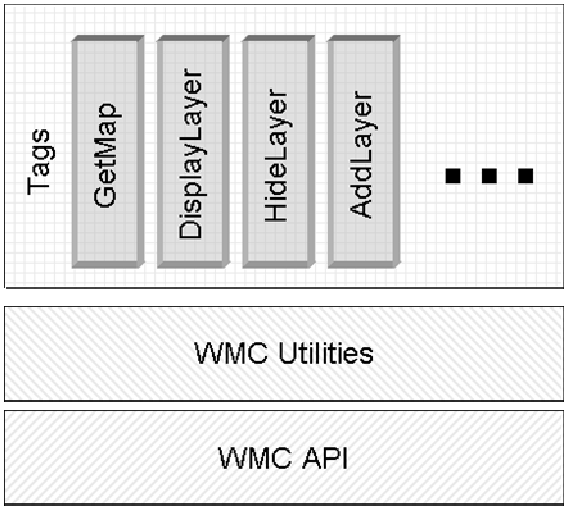


Figure 90 Tags

"ViewContextCollection" documents (Interfaces pp. 130).

Commonly needed functions like move, scale, order, list, display, hide, add, remove, symbolise, generate WMS and WFS requests, etc. are developed based on the WMC Java API. These "Utility" classes interact with OWS services and change the values in the "ViewContext" and / or "ViewContextCollection" document if necessary (Utility classes pp. 151).

According to the state of the art of system architecture (Alur et al. 2003) the business logic is separated from the view. The author used for this "Java Server Pages Tags". Technical details about this technology go beyond the scope of the thesis and can be found on the following website: http://java.sun.com/jsp.

Basically, tags foster the reuse of code and avoid scripting inside web pages. Consequently, tags for the needed functionality were developed by the author.

As shown in the following concrete example (Figure 91 and Figure 92) the use of tags significantly improves the readability and maintainability of web programming. A detailed description of the developed tags can be found in the Appendix (pp. 139).

```
<%
net.opengis.contextViewContext = viewContext
        (net.opengis.context.ViewContext)pageContext.findAttribute(“viewContext”);
com.hadrbolec.ogc.context.dto.MapDTO map
        =com.hadrbolec.ogc.context.render.RenderFactory.renderMap(viewContext);
StringBuffer sb = new StringBuffer(“<div style="position:relative;width:”);
sb.append((map.window.width+6)).append(“px;”);
sb.append(“height:”)append((map.window.height+6)).append(“px;”);
sb.append(“border-style:outset;border-width:3px\">”);
for (int i=0:i<map.getImages().size();i++) {
   sb.append(“ <img style="position:absolute;top:0px;left:0px" src=\””);
   sb.append(((com.hadrbolec.ogc.context.dto.Image)map.getImages().get(i)).getURI()).append(“\””);
   sb.append(“ width=\””).append(map.getWindow().getWidth()").append(“\””);
   sb.append(“ height=\””).append(map.getWindow().getHeight()").append(“\””);
   sb.append(“/>\n”);
}
sb.append(“</div>\n”);
out.println(sb.toString());
%>
```

Figure 91 WMS “GetMap” request: with utility classes in a jsp page.

```
<context:GetMap viewContext="The ViewContext"/>
```

Figure 92 WMS “GetMap” request: with tag file in a jsp page.

4.3 Outcome

The result is an easy to use “Tag library” that is able to perform the necessary GIS functionality identified in the “Establishment of requirements” chapter (pp. 19). It is used for the the following practical review.

5 Practical review

5.1 Test setup

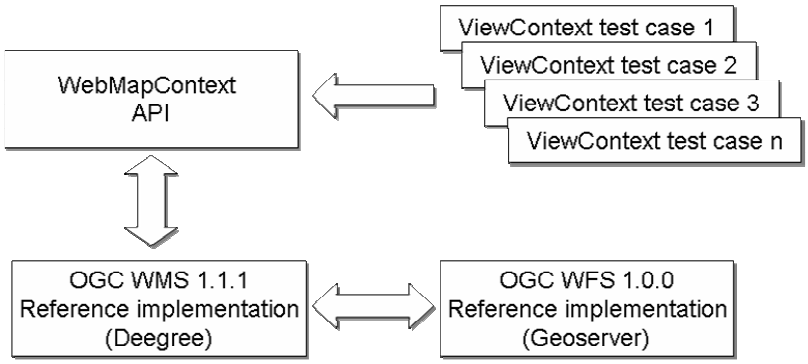


Figure 93 Proof of concept setup

The required functions are assessed with test cases which are set up in "ViewContext" documents (see Appendix "ViewContext documents" pp. 155). The relevant "ViewContext" document is parsed with the author's "WebMap Context" implementation (Implementation p. 65). The functions are tested against the OGC WMS 1.1.1 reference implementation ("Deegree" software). In case that a remote WFS server is needed the OGC WFS reference implementation ("GeoServer" project) is used. The applied datasets are delivered with the reference implementations of the Deegree WMS and the Geoserver WFS.

5.2 Basic system capabilities

5.2.1 Data input

5.2.1.1 Load WMC file (deserialize)

This test starts with loading and parsing of a "ViewContext" file. It does not matter whether the file is retrieved from the local file system or from a "Uniform Resource Locator" (URL). The same approach works with a "ViewContextCollection" document as well. The resulting maps are displayed in Figure 95.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="context" tagdir="/WEB-INF/tags/context" %>
<html>
   <head><title>Load debug file</title></head>
   <body>
      <h2>Load debug file</h2>
      <h3>Load ViewContext document from a file</h3>
      <context:ViewContext
     file="D:/SoftwareDevelopment/WebMapProofOfConcept/web/debug/wmc/WMS_cite_WMC.xml">
         <context:GetMap viewContext="${viewContext}"/>
      </context:ViewContext>

      <h3>Load ViewContext document from an URL</h3>
      <context:ViewContext
         url="http://127.0.0.1:8084/deegreewms/debug/wmc/WMS_cite_WMC.xml">
         <context:GetMap viewContext="${viewContext}"/>
      </context:ViewContext>
   </body>
</html>
```

Figure 94 Jsp file: Load.jsp

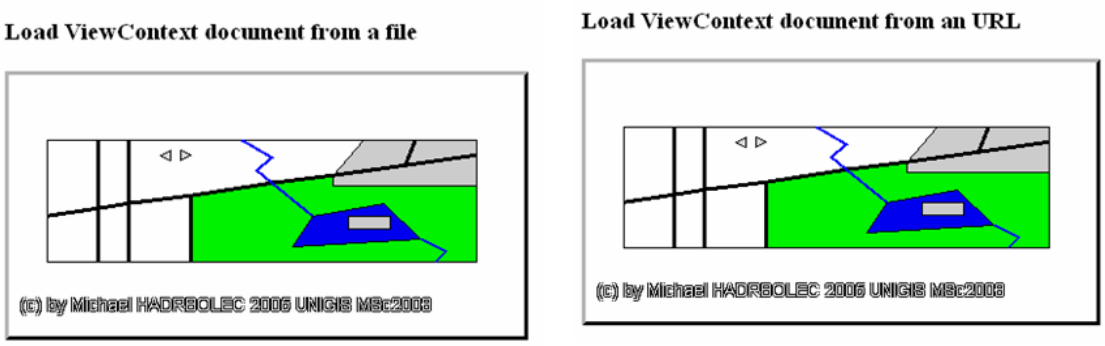


Figure 95 Browser output: Load.jsp

5.2.1.2 Meta-information & Exchange / Teamwork

These topics are discussed and examples are shown in detail in the theoretic review of the specification (pp. 31).

5.2.2 Storing, maintaining, and outputting data

5.2.2.1 Save WMC file (serialize /store) for later usage

During this test the "WMS_cite_WMC.xml" view context document (Appendix: ViewContext documents pp. 155) is saved. The file can be saved into the local file system with the "ViewContextWriter" tag (pp. 139). To assess if this step was done accurately, the serialised file"D:/TestViewContext.xml"is loaded and subsequently displayed. The expected output is the same map as derived from the URL.

```
<%@page contentType="text/html" pageEncoding="UTF-8" %>
<%@taglib prefix="context" tagdir="/WEB-INF/tags/context" %>
<html>
  <head><title>Load debug file</title></head>
  <body>
    <h2>Load debug file</h2>
    <h3>Load ViewContext document from an URL</h3>
    <context:ViewContext
          url="http://127.0.0.1:8084/deegreewms/debug/wmc/WMS_cite_WMC.xml">
      <context:GetMap viewContext="${viewContext}"/>
      <context:ViewContextWriter file="D:/TestViewContext.xml" viewContext="${viewContext}"/>
    </context:ViewContext>
    <h3>Load serialized ViewContext from filesystem</h3>
    <context:ViewContext file="D:/TestViewContext.xml">
      <context:GetMap viewContext="${viewContext}"/>
    </context:ViewContext>
  </body>
</html>
```

Figure 96 Example: Write "ViewContext" document into file system

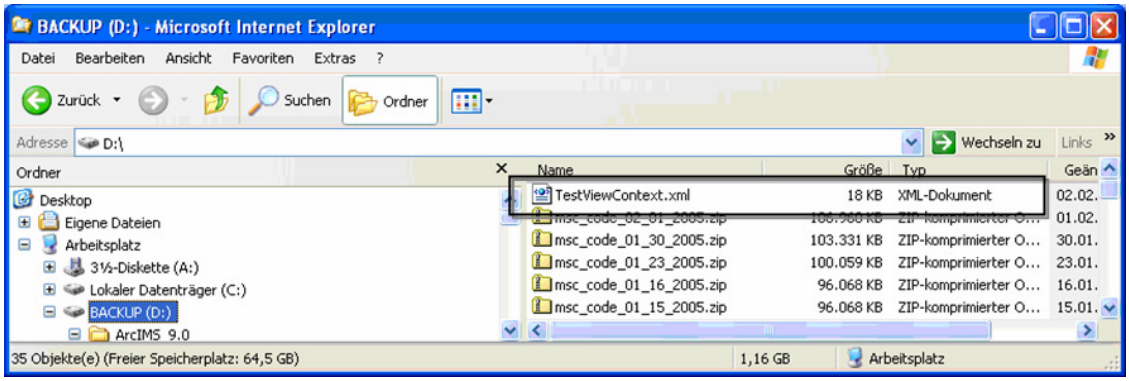


Figure 97 Resulting file in file system

This XML file can be reused later on by the expert who saved it. Another possibility is to mail it to another expert for information exchange and /or teamwork.

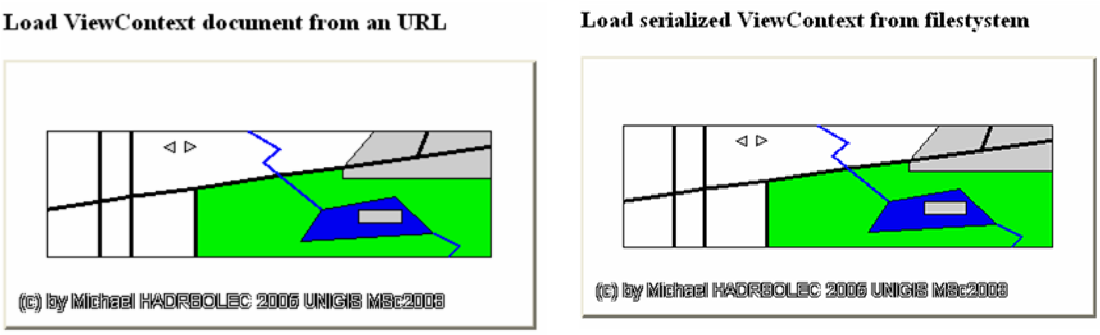


Figure 98 Browser output: Serialize.jsp

5.2.2.2 Edit and display (on output)

5.2.2.3 Handling multiple views

Handling multiple views is nothing else than dealing with multiple "ViewContext" documents. This process needs the same reading and writing mechanism as for the "ViewContext" document. This is the reason, why it is not evaluated in detail here.

5.2.2.4 Visibility

For the visibility test the "WMS_cite_WMC.xml" (Appendix: ViewContext documents pp. 155) view context document is used. First the layers "cite:Lakes" and "cite:NamedPlaces" are hidden with the help of the "HideLayer" tag (Appendix: Tag files p. 140). This is achieved by setting the hidden attribute of their "Layer" element to "true". The resulting map is displayed. In the next step, the layers are displayed with the "DisplayLayer" tag (Appendix: Tag files p. 140), by setting the hidden attribute of their "Layer" element to "false". Once more the map is displayed. The expected output is the same map as derived from the URL.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="context" tagdir="/WEB-INF/tags/context" %>
<html>
  <head><title>Hide and display layer debug file</title></head>
  <body>
    <context:ViewContext
      url="http://127.0.0.1:8084/deegreewms/debug/wmc/WMS_cite_WMC.xml">
      <h3>Original map</h3>
      <context:GetMap viewContext="${viewContext}"/>

      <h3>Hide two layers (cite:NamedPlaces, cite:Lakes)</h3>
      <context:HideLayer layer="cite:Lakes" viewContext="${viewContext}"/>
      <context:HideLayer layer="cite:NamedPlaces" viewContext="${viewContext}"/>
      <context:GetMap viewContext="${viewContext}"/>
```

```
        <h3>Display the two layers (cite:NamedPlaces, cite:Lakes)</h3>
        <context:DisplayLayer layer="cite:Lakes" viewContext="${viewContext}"/>
        <context:DisplayLayer layer="cite:NamedPlaces" viewContext="${viewContext}"/>
        <context:GetMap viewContext="${viewContext}"/>
      </context:ViewContext>
   </body>
</html>
```

Figure 99 Jsp file: LayerVisibility.jsp

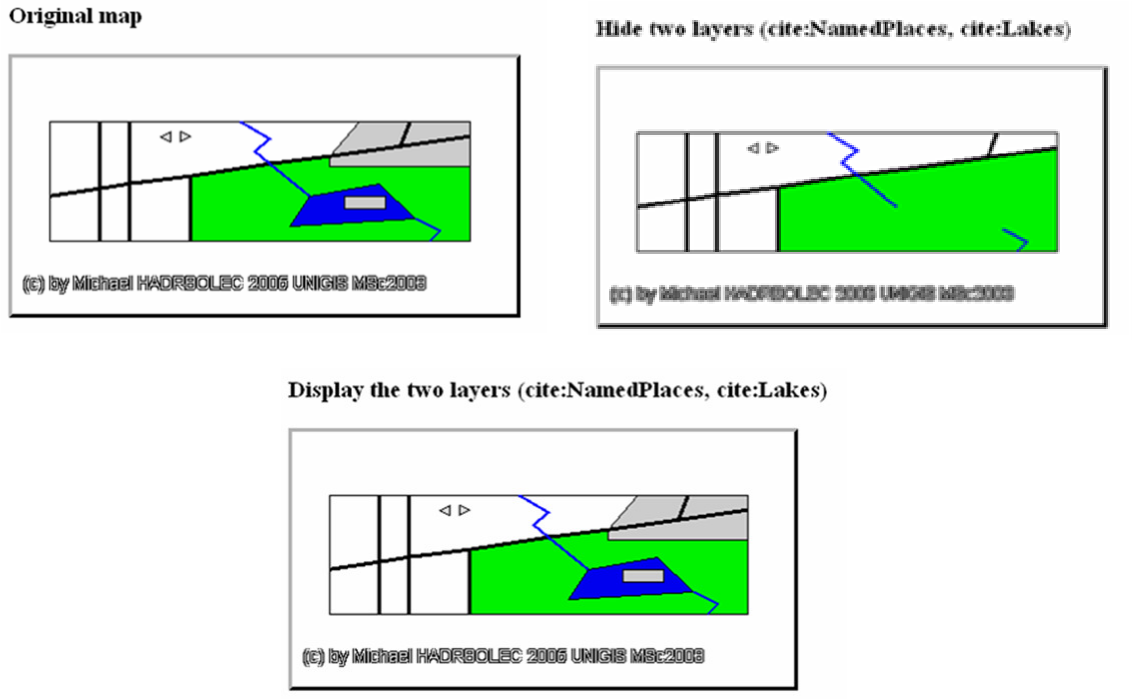


Figure 100 Browser output: LayerVisibility.jsp

5.2.2.5 Add layer

The test is combinded with Annotation / highlighting, see page 80 for more details.

5.2.2.6 Remove layer

Once more a view context document is used ("WMS_cite_WMC.xml" Appendix: ViewContext documents pp. 155). In this test case the "cite:Lakes" and the "cite:NamedPlaces" layers are completely removed from the "LayerList" element with the "RemoveLayer" tag (Appendix: Tag files pp. 139). The expected output is a map without both layers.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="context" tagdir="/WEB-INF/tags/context" %>
<html>
   <head><title>Remove layer debug file</title></head>
   <body>
```

```
    <context:ViewContext
        url="http://127.0.0.1:8084/deegreewms/debug/wmc/WMS_cite_WMC.xml">
      <h3>Original map</h3>
      <context:GetMap viewContext="${viewContext}"/>

      <h3>Remove "cite:lakes" and "cite:Namedaplaces" layer</h3>
      <context:RemoveLayer layerName="cite:Lakes" viewContext="${viewContext}"/>
      <context:RemoveLayer layerName="cite:NamedPlaces" viewContext="${viewContext}"/>
      <context:GetMap viewContext="${viewContext}"/>
    </context:ViewContext>
  </body>
</html>
```

Figure 101 Jsp file: Remove.jsp

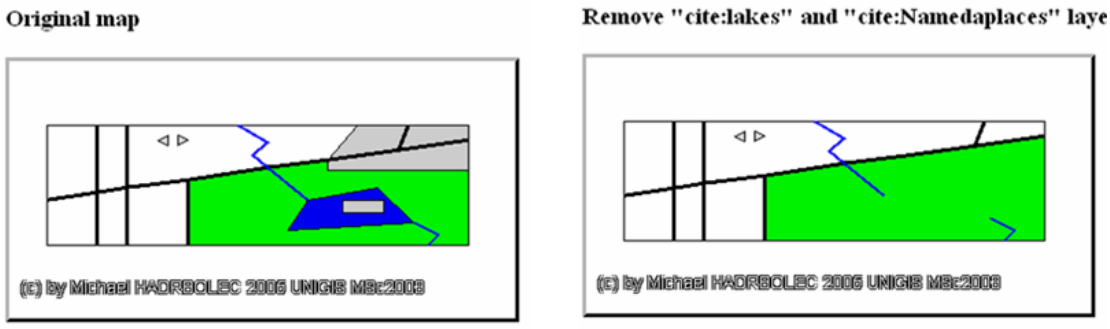


Figure 102 Browser output: Remove.jsp

5.2.2.7 Order layer

For this test the "WMS_cite_WMC.xml" (Appendix: ViewContext documents pp. 155) view context document is used again. The order of the layers can be simply changed by rearranging the "Layer" elements in the "LayerList". In this case the layer "cite:NamedPlaces" is moved to the last position of all layer elements in the "LayerList" (bottom of the list) with the "OrderLayer" tag (Appendix: Tag files pp. 142). This has the effect that this layer is displayed on top position in the map.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="context" tagdir="/WEB-INF/tags/context" %>
<html>
  <head><title>Order layer debug file</title></head>
  <body>
    <h2>Order layer debug file</h2>
    <h3>Original layer order </h3>
    <context:ViewContext
        url="http://127.0.0.1:8084/deegreewms/debug/wmc/WMS_cite_WMC.xml">
      <context:GetMap viewContext="${viewContext}"/>

      <h3>Move "cite:NamedPlaces" top</h3>
      <context:OrderLayer layerName="cite:NamedPlaces" action="top"
```

```
            viewContext="${viewContext}"/>
          <context:GetMap viewContext="${viewContext}"/>
        </context:ViewContext>
    </body>
</html>
```

Figure 103 Jsp file: LayerOrder.jsp

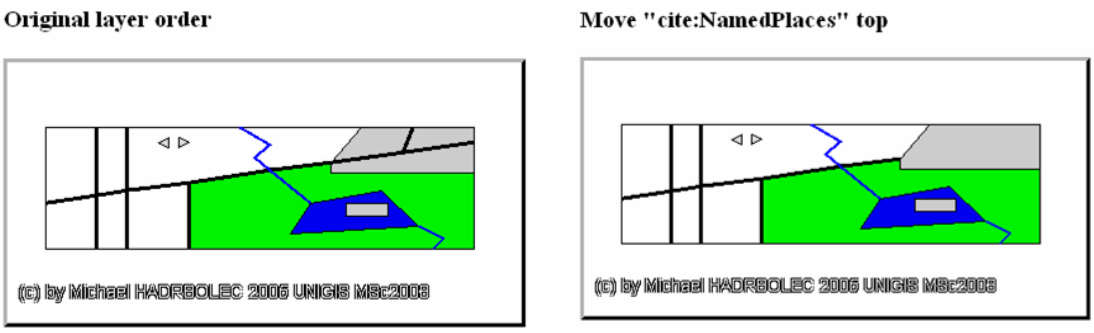


Figure 104 Browser output: LayerOrder.jsp

5.2.2.8 Symbolize

For this scenario the "Style.xml" view context file is used. Symbolisation can be done in different ways. First, a predefined style is used. With the "StyleLayer" tag (Appendix: Tag files pp. 143) the predefined style "default:cite:BasicPolygons" of the layer "cite:BasicPolygons" is selected. The results are solid filled grey polygons which are the same as originally defined in the "Deegree" web map server reference implementation. In the next step the user-defined symbolisation possibilities are assessed. Therefore, an online accessible SLD file "BasicPolygons.sld" is referenced with the "AddStyle" tag (Appendix: Tag files pp. 143). The next steps are adding a "Styled Layer Descriptor" defined inline. Finally, an inline defined "FeatureTypeStyle" is added. As expected black, green and red outlined polygons are subsequently displayed.

```
<?xml version="1.0" encoding="UTF-8"?>
<ViewContext xmlns="http://www.opengis.net/context" xmlns:sld="http://www.opengis.net/sld"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ogc="http://www.opengis.net/ogc"
version="1.0.0" id="U1051_Style">
  <General>
    <Window width="400" height="200"/>
    <BoundingBox SRS="EPSG:4326" minx="-10" miny="-10" maxx="10" maxy="10"/>
  </General>
  <LayerList>
    <Layer queryable="true" hidden="false">
      <Server service="OGC:WMS" version="1.1.1">
        <OnlineResource xlink:type="simple" xlink:href="http://127.0.0.1:8084/deegreewms/wms?"/>
      </Server>
      <Name>cite:BasicPolygons</Name>
```

```
        <Title>BASIC Polygons</Title>
        <SRS>EPSG:4326</SRS>
        <FormatList>
          <Format current="true">image/gif</Format>
        </FormatList>
        <StyleList>
          <Style current="true">
            <Name>default:cite:BasicPolygons</Name>
            <Title>default:cite:BasicPolygons</Title>
          </Style>
        </StyleList>
      </Layer>
    </LayerList>
</ViewContext>
```

Figure 105 ViewContext file: Style.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<StyledLayerDescriptor version="String" xmlns="http://www.opengis.net/sld"
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <NamedLayer>
    <Name>cite:BasicPolygons</Name>
    <UserStyle>
      <Name>default:cite:BasicPolygons</Name>
      <Title>Userdefined</Title>
      <FeatureTypeStyle>
        <Name>default:cite:BasicPolygons</Name>
        <Rule>
          <Name>cite:BasicPolygons</Name>
          <PolygonSymbolizer>
            <Geometry>
              <ogc:PropertyName>GEOM</ogc:PropertyName>
            </Geometry>
            <Stroke>
              <CssParameter name="stroke">#000000</CssParameter>
              <CssParameter name="stroke-width">3</CssParameter>
            </Stroke>
          </PolygonSymbolizer>
        </Rule>
      </FeatureTypeStyle>
    </UserStyle>
  </NamedLayer>
</StyledLayerDescriptor>
```

Figure 106 Online referenced SLD file: BasicPolygons.sld

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="context" tagdir="/WEB-INF/tags/context" %>
<html>
  <head><title>Symbolize debug file</title></head>
  <body>
    <h2>User defined symbolise debug file</h2>
    <context:ViewContext url=" http://localhost:8084/deegreewms/debug/style/Style.xml">
      <h3>Predefined style</h3>
      <context:StyleLayer layerName="cite:BasicPolygons" styleName="default:cite:BasicPolygons"
        viewContext="${viewContext}"/>
      <context:GetMap viewContext="${viewContext}"/>
      <h3>User defined online resource</h3>
      <context:AddStyle layerName="cite:BasicPolygons" viewContext="${viewContext}">
        <Style>
          <SLD>
            <Name>online:styledLayeDescriptorFile:cite:BasicPolygons</Name>
            <Title>Online basic polygon style</Title>
            <OnlineResource xlink:type="simple"
              xlink:href="http://127.0.0.1:8084/deegreewms/BasicPolygons.sld"/>
          </SLD>
        </Style>
      </context:AddStyle>
      <context:GetMap viewContext="${viewContext}"/>
      <h3>User defined inline "StyledLayerDescriptor" resource</h3>
      <context:AddStyle layerName="cite:BasicPolygons" viewContext="${viewContext}">
        <Style>
        <SLD>
          <Name>inline:styledLayeDescriptor:cite:BasicPolygons</Name>
          <Title>Inline SLD basic polygon style</Title>
          <sld:StyledLayerDescriptor version="1.0.0">
            <sld:NamedLayer>
              <sld:Name>cite:BasicPolygons</sld:Name>
              <sld:UserStyle>
                <sld:Name>default:cite:BasicPolygons</sld:Name>
                <sld:Title>Userdefined</sld:Title>
                <sld:FeatureTypeStyle>
                  <sld:Name>default:cite:BasicPolygons</sld:Name>
                  <sld:Rule>
                    <sld:Name>cite:BasicPolygons</sld:Name>
                    <sld:PolygonSymbolizer>
                      <sld:Geometry>
                        <ogc:PropertyName>GEOM</ogc:PropertyName>
                      </sld:Geometry>
                      <sld:Stroke>
                        <sld:CssParameter name="stroke">#00FF00</sld:CssParameter>
```

```
                                <sld:CssParameter name="stroke-width">5</sld:CssParameter>
                            </sld:Stroke>
                        </sld:PolygonSymbolizer>
                    </sld:Rule>
                </sld:FeatureTypeStyle>
            </sld:UserStyle>
        </sld:NamedLayer>
    </sld:StyledLayerDescriptor>
</SLD>
</Style>
```

```
</context:AddStyle>
<context:GetMap viewContext="${viewContext}"/>
<h3>User defined inline "FeatureTypeStyle" resource</h3>
<context:AddStyle layerName="cite:BasicPolygons" viewContext="${viewContext}">
```

```
<Style>
<SLD>
    <Name>inline:featureTypeStyle:cite:BasicPolygons</Name>
    <Title>inline feature type style basic polygon style</Title>
    <sld:FeatureTypeStyle>
        <sld:Name>default:cite:BasicPolygons</sld:Name>
        <sld:Rule>
            <sld:Name>cite:BasicPolygons</sld:Name>
            <sld:PolygonSymbolizer>
                <sld:Geometry>
                    <ogc:PropertyName>GEOM</ogc:PropertyName>
                </sld:Geometry>
                <sld:Stroke>
                    <sld:CssParameter name="stroke">#FF0000</sld:CssParameter>
                    <sld:CssParameter name="stroke-width">7</sld:CssParameter>
                </sld:Stroke>
            </sld:PolygonSymbolizer>
        </sld:Rule>
    </sld:FeatureTypeStyle>
</SLD>
</Style>
```

```
</context:AddStyle>
<context:GetMap viewContext="${viewContext}"/>
</context:ViewContext>
</body>
</html>
```

Figure 107 Jsp: UserDefinedSymbolisation.jsp

```
<div style="position:relative;width:406px;height:206px;border-style:outset;border-width:3px">
<img style="position:absolute;top:0px;left:0px"
src="http://127.0.0.1:8084/deegreewms/wms?SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&BBox=-10.0%2C-10.0%2C10.0%2C10.0&SRS=EPSG%3A4326&FORMAT=image%2Fgif&width=400&height=200&TRANSPARENT=true&LAYERS=cite%3ABasicPolygons&STYLES=default%3Acite%3ABasicPolygons&" width="400" width="200"/>
</div>
```

Figure 108 HTML fragment: predefined symbolisation

```
<img style="position:absolute;top:0px;left:0px"
src="http://127.0.0.1:8084/deegreewms/wms?SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&BBox=-10.0%2C-10.0%2C10.0%2C10.0&SRS=EPSG%3A4326&FORMAT=image%2Fgif&width=400&height=200&TRANSPARENT=true&SLD=http%3A%2F%2F127.0.0.1%3A8084%2Fdeegreewms%2FBasicPolygons.sld&" width="400" width="200"/>
```

Figure 109 HTML output fragment: Online accesssable SLD

If a user-defined style is submitted with a HTTP GET request it has to be encoded and provided as SLD_BODY key value pair (et. al Lalonde pp. 7-9). This operation is carried out with the help of the author's utility classes. Unfortunately, the length of the browsers cache is limited. The result is that in some cases only a request fragment (= invalid request) is submitted to the WMS server. An alternative would be to submit a HTTP POST request. However, according to Beaujardiere (2002, p. 13) *"[…] requests using HTTP POST have not yet been defined for a basic Web Map Service"*.

```
<img style="position:absolute;top:0px;left:0px"
src="http://127.0.0.1:8084/deegreewms/wms?SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&BBox=-10.0%2C-10.0%2C10.0%2C10.0&SRS=EPSG%3A4326&FORMAT=image%2Fgif&width=400&height=200&TRANSPARENT=true&
SLD_BODY=%3C%3Fxml+version%3D%221.0%22+%3F%3E%3CStyledLayerDescriptor+xmlns%3D%22http%3A%2F%2Fwww.opengis.net%2Fsld%22+xmlns%3Asld%3D%22http%3A%2F%2Fwww.opengis.net%2Fsld%22+xmlns%3Aogc%3D%22http%3A%2F%2Fwww.opengis.net%2Fogc%22+xmlns%3Agml%3D%22http%3A%2F%2Fwww.opengis.net%2Fgml%22+xmlns%3Axlink%3D%22http%3A%2F%2Fwww.w3.org%2F1999%2Fxlink%22+version%3D%221.0.0%22%3E%3Csld%3ANamedLayer%3E%3Csld%3AName%3Ecite%3ABasicPolygons%3C%2Fsld%3AName%3E%3Csld%3AUserStyle%3E%3Csld%3AName%3Edefault%3Acite%3ABasicPolygons%3C%2Fsld%3AName%3E%3Csld%3ATitle%3EUserdefined%3C%2Fsld%3ATitle%3E%3Csld%3AFeatureTypeStyle%3E%3Csld%3AName%3Edefault%3Acite%3ABasicPolygons%3C%2Fsld%3AName%3E%3Csld%3ARule%3E%3Csld%3AName%3Ecite%3ABasicPolygons%3C%2Fsld%3AName%3E%3Csld%3APolygonSymbolizer%3E%3Csld%3AGeometry%3E%3Cogc%3APropertyName%3EGEOM%3C%2Fogc%3APropertyName%3E%3C%2Fsld%3AGeometry%3E%3Csld%3AStroke%3E%3Csld%3ACssParameter+name%3D%22stroke%22%3E%2300FF00%3C%2Fsld%3ACssParameter%3E%3Csld%3ACssParameter+name%3D%22stroke-width%22%3E5%3C%2Fsld%3ACssParameter%3E%3C%2Fsld%3AStroke%3E%3C%2Fsld%3APolygonSymbolizer%3E%3C%2Fsld%3ARule%3E%3C%2Fsld%3AFeatureTypeStyle%3E%3C%2Fsld%3AUserStyle%3E%3C%2Fsld%3ANamedLayer%3E%3C%2FStyledLayerDescriptor%3E&
" width="400" width="200"/>
```

Figure 110 Html output fragment: StyledLayerDescriptor

<table>
<tr><td><img style="position:absolute;top:0px;left:0px"
src="http://127.0.0.1:8084/deegreewms/wms?SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&BBox=-10.0%2C-10.0%2C10.0%2C10.0&SRS=EPSG%3A4326&FORMAT=image%2Fgif&width=400&height=200&TRANSPARENT=true&</td></tr>
<tr><td>SLD_BODY=%3C%3Fxml+version%3D%221.0%22+%3F%3E%3CStyledLayerDescriptor+xmlns%3D%22http%3A%2F%2Fwww.opengis.net%2Fsld%22+xmlns%3Asld%3D%22http%3A%2F%2Fwww.opengis.net%2Fsld%22+xmlns%3Aogc%3D%22http%3A%2F%2Fwww.opengis.net%2Fogc%22+xmlns%3Agml%3D%22http%3A%2F%2Fwww.opengis.net%2Fgml%22+xmlns%3Axlink%3D%22http%3A%2F%2Fwww.w3.org%2F1999%2Fxlink%22+version%3D%221.0.0%22%3E%3Csld%3ANamedLayer%3E%3Csld%3AName%3Ecite%3ABasicPolygons%3C%2Fsld%3AName%3E%3Csld%3AUserStyle%3E%3Csld%3AFeatureTypeStyle%3E%3Csld%3AName%3Edefault%3Acite%3ABasicPolygons%3C%2Fsld%3AName%3E%3Csld%3ARule%3E%3Csld%3AName%3Ecite%3ABasicPolygons%3C%2Fsld%3AName%3E%3Csld%3APolygonSymbolizer%3E%3Csld%3AGeometry%3E%3Cogc%3APropertyName%3EGEOM%3C%2Fogc%3APropertyName%3E%3C%2Fsld%3AGeometry%3E%3Csld%3AStroke%3E%3Csld%3ACssParameter+name%3D%22stroke%22%3E%23FF0000%3C%2Fsld%3ACssParameter%3E%3Csld%3ACssParameter+name%3D%22stroke-width%22%3E7%3C%2Fsld%3ACssParameter%3E%3C%2Fsld%3AStroke%3E%3C%2Fsld%3APolygonSymbolizer%3E%3C%2Fsld%3ARule%3E%3C%2Fsld%3AFeatureTypeStyle%3E%3C%2Fsld%3AUserStyle%3E%3C%2Fsld%3ANamedLayer%3E%3C%2FStyledLayerDescriptor%3E&</td></tr>
<tr><td>" width="400" width="200"/></td></tr>
</table>

Figure 111 Html output fragment: FeatureTypeStyle

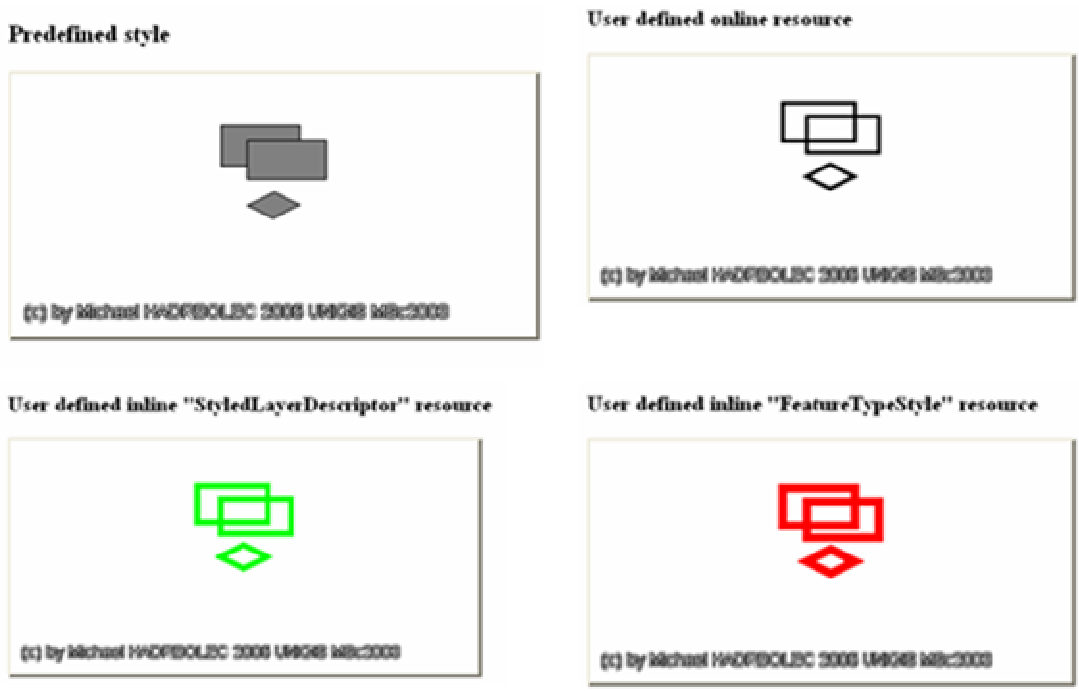


Figure 112 Browser output: UserDefinedSymbolisation.jsp

5.2.2.8.1 Annotation / highlighting

During this test scenario the "WMS_cite_WMC.xml" view context document is used again (Appendix: ViewContext documents pp. 155). The new layer "MyComment", which contains the annotation "my comment" and highlights the "GooseIsland", is added with the "AddLayer" tag (Appendix: Tag files pp. 141) to the layer list. It references the "cite:NamedPlaces" layer.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="context" tagdir="/WEB-INF/tags/context" %>
<html>
  <head><title>Add layer debug file</title></head>
  <body>
    <context:ViewContext
        url="http://127.0.0.1:8084/deegreewms/debug/wmc/WMS_cite_WMC.xml">
      <h3>Original layer</h3>
      <context:GetMap viewContext="${viewContext}">
        <context:DisplayMap map="${map}"/>
      </context:GetMap>
      <h3>Add an annotation layer that comments an highlights</h3>
      <context:AddLayer viewContext="${viewContext}">
        <Layer queryable="true" hidden="false" xmlns="http://www.opengis.net/context"
            xmlns:sld="http://www.opengis.net/sld" xmlns:xlink="http://www.w3.org/1999/xlink"
            xmlns:ogc="http://www.opengis.net/ogc">
          <Server service="OGC:WMS" version="1.1.1">
            <OnlineResource xlink:type="simple"
                xlink:href="http://127.0.0.1:8084/deegreewms/wms?"/>
          </Server>
          <Name>MyComment</Name>
          <Title>My comment</Title>
          <SRS>EPSG:4326</SRS>
          <FormatList>
            <Format current="true">image/gif</Format>
          </FormatList>
          <StyleList>
            <Style current="true">
            <SLD>
              <StyledLayerDescriptor version="1.0.0" xmlns="http://www.opengis.net/sld"
                xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
                xmlns:xlink="http://www.w3.org/1999/xlink">
                <NamedLayer>
                  <Name>cite:NamedPlaces</Name>
                  <UserStyle>
                    <Name>myAnnotation</Name>
                    <Title>My map annotation</Title>
                    <FeatureTypeStyle>
                      <Name>default:cite:NamedPlaces</Name>
                      <Rule>
                        <ogc:Filter>
                          <ogc:PropertyIsEqualTo>
                            <ogc:PropertyName>NAME</ogc:PropertyName>
                            <ogc:Literal>Goose Island</ogc:Literal>
                          </ogc:PropertyIsEqualTo>
                        </ogc:Filter>
                        <PolygonSymbolizer>
```

```
                        <Geometry>
                          <ogc:PropertyName>GEOM</ogc:PropertyName>
                        </Geometry>
                        <Fill>
                          <CssParameter name="fill">#FFFFFF</CssParameter>
                        </Fill>
                        <Stroke>
                          <CssParameter name="stroke">#FF0000</CssParameter>
                          <CssParameter name="stroke-width">3</CssParameter>
                        </Stroke>
                      </PolygonSymbolizer>
                      <TextSymbolizer>
                        <Label>My personal comment</Label>
                        <Font>
                          <CssParameter name="font-family">Arial</CssParameter>
                          <CssParameter name="font-family">Sans-Serif</CssParameter>
                          <CssParameter name="font-style">italic</CssParameter>
                          <CssParameter name="font-size">20</CssParameter>
                          <CssParameter name="font-color">#000000</CssParameter>
                        </Font>
                        <LabelPlacement>
                          <PointPlacement>
                            <Displacement>
                            <DisplacementX>-150</DisplacementX>
                            <DisplacementY>30</DisplacementY>
                            </Displacement>
                          </PointPlacement>
                        </LabelPlacement>
                        <Halo>
                          <Fill>
                            <CssParameter name="fill">#ffff00</CssParameter>
                            <CssParameter name="fill-opacity">0.8</CssParameter>
                          </Fill>
                        </Halo>
                      </TextSymbolizer>
                    </Rule>
                  </FeatureTypeStyle>
                </UserStyle>
              </NamedLayer>
            </StyledLayerDescriptor>
          </SLD>
          </Style>
        </StyleList>
        </Layer>
    </context:AddLayer>
    <context:GetMap viewContext="${viewContext}"/>
  </context:ViewContext>
</body>
```

```
</html>
```

Figure 113 Jsp file: AddLayer.jsp

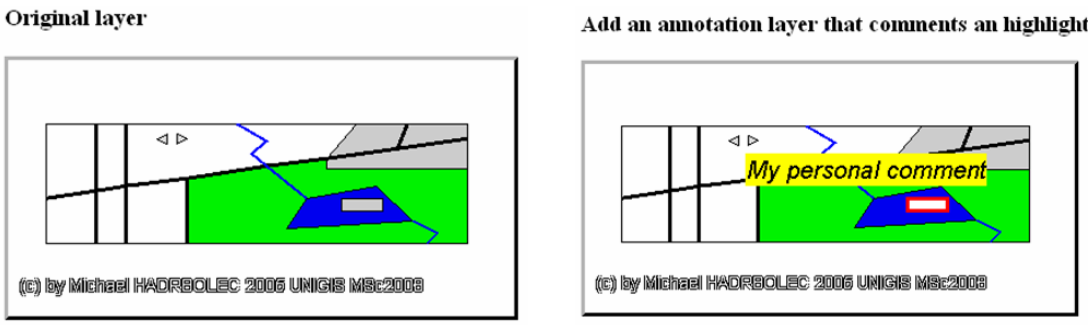


Figure 114 Browser output: AddLayer.jsp

5.2.2.9 Plot

The plot test case demonstrates that the following HTML fragment is printed out with 10 cm width and 7 cm height.

```
<img src=”temp.gif” border=”1” style=”width:10cm;height:7cm;”/>
```

Figure 115 HTML fragment: Print width and height in exact value

The test was successful.

5.2.2.10 Browse

5.2.2.10.1 Move

5.2.2.10.1.1 Move dx, dy

Once more the “WMS_cite_WMC.xml” (Appendix: ViewContext documents pp. 155) view context document is used. During the test the visible map extent moves 200 pixel westwards and 100 pixels northwards with the help of the “Move” tag (Appendix: Tag files pp. 145). In the next step the map is moved back to its original position.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="context" tagdir="/WEB-INF/tags/context" %>
<html>
  <head><title>Move dx, dy debug file</title></head>
  <body>
    <h2>Move dx, dy</h2>
    <context:ViewContext
        url="http://localhost:8084/deegreewms/debug/wmc/WMS_cite_WMC.xml">
      <h3>Original map</h3>
      <context:GetMap viewContext="${viewContext}"/>

      <h3>Move dx="200" dy="100"</h3>
      <context:Move dx="200" dy="100" viewContext="${viewContext}"/>
```

```
        <context:GetMap viewContext="${viewContext}"/>

        <h3>Move dx="-200" dy="-100" (=original map)</h3>
        <context:Move dx="-200" dy="-100" viewContext="${viewContext}"/>
        <context:GetMap viewContext="${viewContext}"/>
    </context:ViewContext>
  </body>
</html>
```

Figure 116 Jsp file: MoveDxDy.jsp

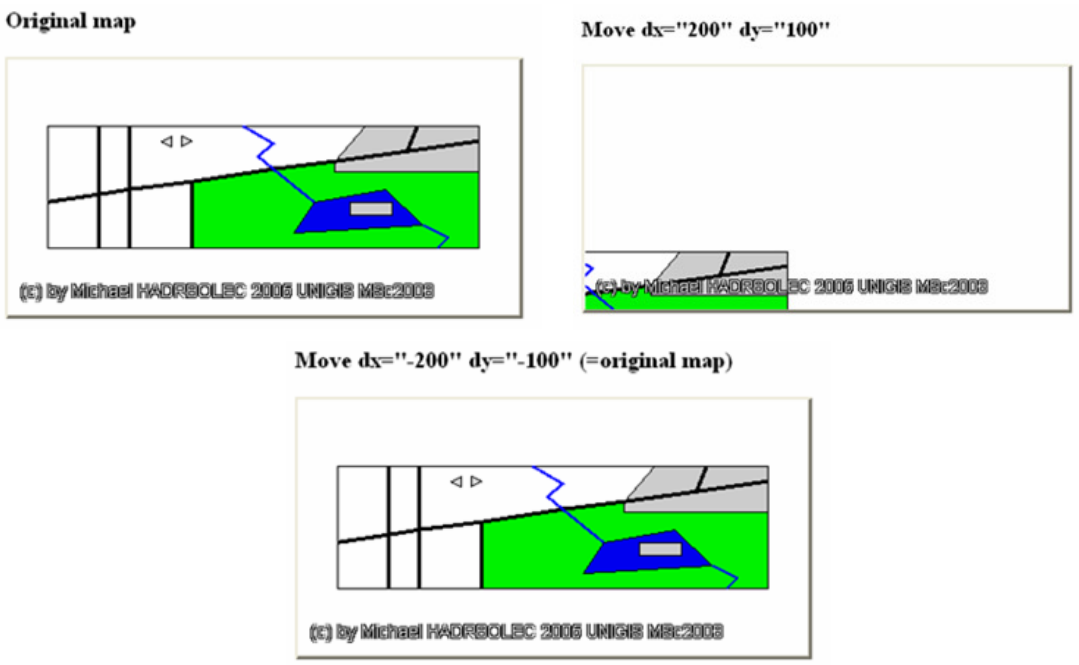


Figure 117 Browser output: MoveDxDy.jsp

5.2.2.10.1.2 Move from point to point (pan)

Now the "WMS_cite_WMC.xml" (Appendix: ViewContext documents pp. 155) view context document is used to move (pan) the visible map extent from the starting point (100, 100) to the end point (300, 150) applying the "Move" tag (Appendix: Tag files pp. 145). In the next step the extent is moved back to the original map extent.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="context" tagdir="/WEB-INF/tags/context" %>
<html>
  <head><title>Move from point to point debug file</title></head>
  <body>
    <h2>Move from point to point</h2>
    <context:ViewContext
        url="http://localhost:8084/deegreewms/debug/wmc/WMS_cite_WMC.xml">

        <h3>Original map</h3>
        <context:GetMap viewContext="${viewContext}"/>

        <h3>Move from point x="100" y="100" to point x="300" y="150"</h3>
```

```
        <context:Move fromX="100" fromY="100" toX="300" toY="150"
            viewContext="${viewContext}"/>
        <context:GetMap viewContext="${viewContext}"/>

        <h3>Move from point x="350" y="175" to point x="150" y="125" (=original map)</h3>
        <context:Move fromX="350" fromY="175" toX="150" toY="125"
            viewContext="${viewContext}"/>
        <context:GetMap viewContext="${viewContext}"/>
    </context:ViewContext>
  </body>
</html>
```

Figure 118 Jsp file: MoveFromPointToPoint.jsp

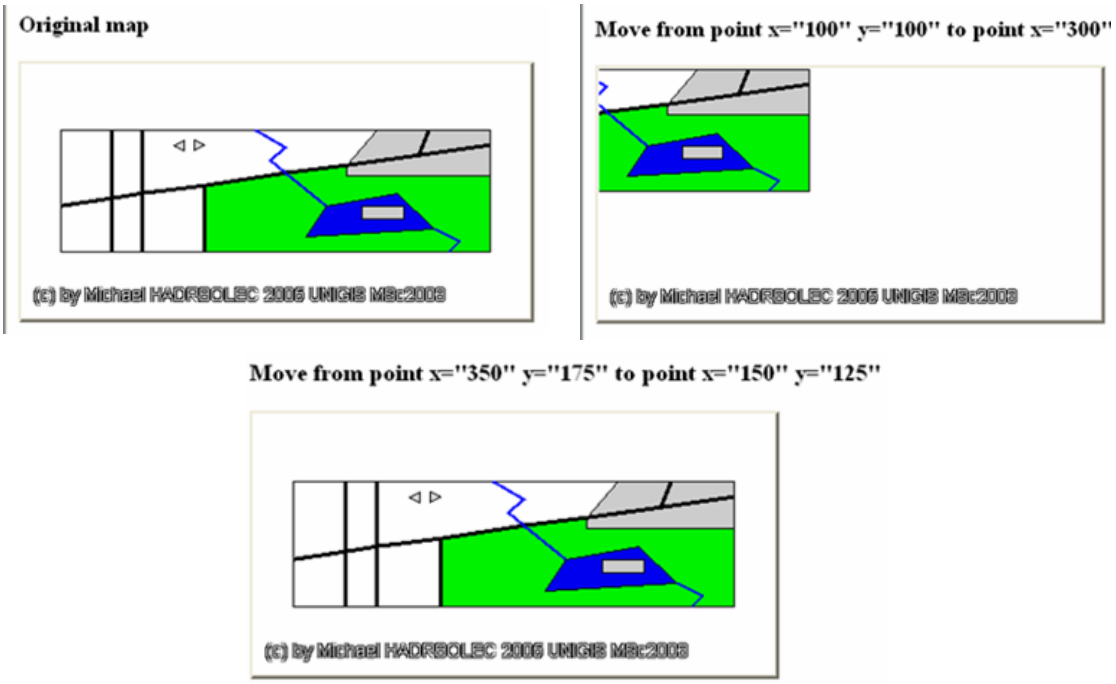


Figure 119 Browser output: MoveFromPointToPoint.jsp

5.2.2.10.1.3 Move to point (centre at)

Once more the "WMS_cite_WMC.xml" (Appendix: ViewContext documents pp. 155) view context document is used. At the beginning the visible map extent is re-centred at the point (300, 150). As next step it is re-centred back at the original centre point (100, 50). This action is performed with the "Move" tag (Appendix: Tag files pp. 139). The expected result is the original map.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="context" tagdir="/WEB-INF/tags/context" %>
<html>
  <head><title>Move to point debug file</title></head>
  <body>
    <h2>Move to point</h2>
```

```
        <context:ViewContext
            url="http://localhost:8084/deegreewms/debug/wmc/WMS_cite_WMC.xml">
          <h3>Original map</h3>
          <context:GetMap viewContext="${viewContext}"/>

          <h3>Move the map to point x="300" y="150"</h3>
          <context:Move toX="300" toY="150" viewContext="${viewContext}"/>
          <context:GetMap viewContext="${viewContext}"/>

          <h3>Move the map to point x="100" y="50" (=original map)</h3>
          <context:Move toX="100" toY="50" viewContext="${viewContext}"/>
          <context:GetMap viewContext="${viewContext}"/>
        </context:ViewContext>
    </body>
</html>
```

Figure 120 Jsp file: MoveToPoint.jsp

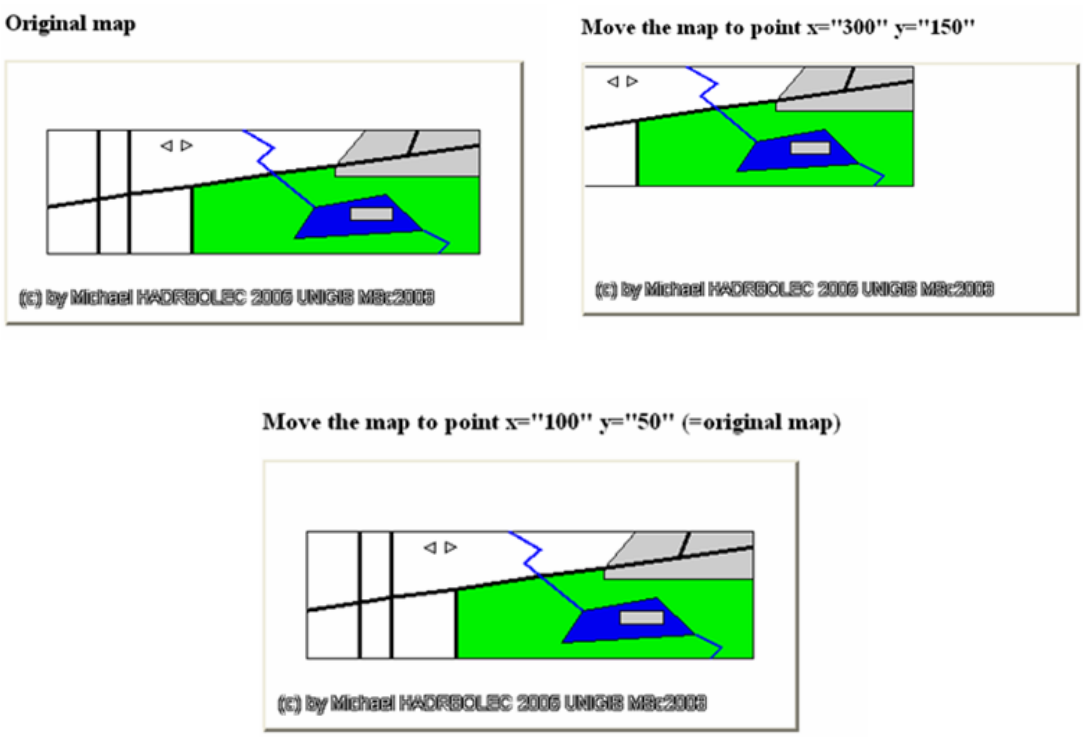


Figure 121 Browser output: MoveToPoint.jsp

5.2.2.10.2 Identify

Again, the "WMS_cite_WMC.xml" view context document is used (Appendix: ViewContext documents pp. 155). This test performs an "GetFeatureInfo" request on the screen for the "cite:NamedPlaces" layer with the "GetFeatureInfo" tag (Appendix: Tag files pp. 146). The expected result is the "GetFeatureInfo" response from the WMS server for the named place "Ashton".

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@taglib prefix="context" tagdir="/WEB-INF/tags/context" %>
<html>
  <head><title>GetFeatureInfo debug file</title></head>
  <body>
    <h2>GetFeatureInfo </h2>
    <context:ViewContext
       url="htpp://127.0.0.1:8084/deegreewms/debug/wmc/WMS_cite_WMC.xml">
      <h3>Original map</h3>
      <context:GetMap viewContext="${viewContext}"/>
      <h3>GetFeatureInfo request (x=300, y=75 layerName="cite:NamedPlaces")</h3>
      <context:GetFeatureInfo x="300" y="75" layerName="cite:NamedPlaces"
        viewContext="${viewContext}">
        <b>${request}</b>
        <c:out value="${response}" escapeXml="true"/>
      </context:GetFeatureInfo>
    </context:ViewContext>
  </body>
</html>
```

Figure 122 Jsp file: WMS GetFeatureInfo request

```
http://127.0.0.1:8084/deegreewms/wms?SERVICE=WMS&VERSION=1.1.1&LAYERS=cite%3ANamedPlaces&REQUEST=GetFeatureInfo&BBox=-0.0050%2C-0.0050%2C0.0050%2C0.0050&SRS=EPSG%3A4326&FORMAT=image%2Fgif&width=400&height=200&TRANSPARENT=true&QUERY_LAYERS=cite%3ANamedPlaces&X=300&Y=75&
```

Figure 123 WMS GetFeatureInfo request

```
<?xml version="1.0" encoding="UTF-8"?>
<ll:FeatureCollection xmlns:gml="http://www.opengis.net/gml" xmlns:ll="http://www.lat-lon.de"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <gml:boundedBy>
    <gml:Box srsName="EPSG:4326">
      <gml:coordinates>-0.0050,-0.0050 0.0050,0.0050</gml:coordinates>
    </gml:Box>
  </gml:boundedBy>
  <gml:featureMember>
    <ll:NamedPlaces fid="Ashton">
      <ll:FID>117</ll:FID>
      <ll:NAME>Ashton</ll:NAME>
    </ll:NamedPlaces>
  </gml:featureMember>
</ll:FeatureCollection>
```

Figure 124 WMS GetFeatureInfo request response

This XML response can easily be further transformed with a style sheet (Create list (report) pp. 88).

5.2.2.11 Suppress

See “Query” (pp. 92) for more details. The same mechanisms are applied in both cases.

5.2.2.12 Create list (report)

For this test the “ListReport_WMC.xml” view context file is used. It defines an “ogc:Filter” element in the “LayerFeatureConstraints” section. Only features of the “topp:states” layer should be listed and displayed if the property “topp:STATE_NAME” equals “California”. With the help of the “GetFeature” tag (p. 147) in the “List.jsp” JSP file a WFS request is created from the “ViewContext” file. The response is displayed and later on transformed with the”List.xsl” XSL stylesheet into a “Hypertext Markup Language” (HTML) page.

```
<?xml version="1.0" encoding="UTF-8"?>
<ViewContext xmlns="http://www.opengis.net/context" xmlns:sld="http://www.opengis.net/sld"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.0.0" id="U1051_Cite">
  <General>
    <Window width="400" height="200"/>
    <BoundingBox SRS="EPSG:4326" minx="-126" miny="18" maxx="-66" maxy="57"/>
    <Title>Layer order 1</Title>
  </General>
  <LayerList>
    <Layer queryable="true" hidden="false">
    <Server service="OGC:WMS" version="1.1.1">
      <OnlineResource xlink:type="simple" xlink:href="http://127.0.0.1:8084/deegreewms/wms?"/>
    </Server>
    <Name>States</Name>
    <Title>States</Title>
    <SRS>EPSG:4326</SRS>
    <FormatList>
      <Format current="true">image/gif</Format>
    </FormatList>
    <StyleList>
      <Style current="true">
      <SLD>
        <Name>Original</Name>
        <sld:StyledLayerDescriptor version="1.0.0">
          <sld:UserLayer>
            <sld:RemoteOWS>
              <sld:Service>WFS</sld:Service>
              <sld:OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
                  xlink:type="simple" xlink:href="http://127.0.0.1:8084/geoserver/wfs?"/>
            </sld:RemoteOWS>
            <sld:LayerFeatureConstraints>
              <sld:FeatureTypeConstraint>
```

```
                    <sld:FeatureTypeName>topp:states</sld:FeatureTypeName>
                    <ogc:Filter>
                      <ogc:PropertyIsEqualTo>
                        <ogc:PropertyName>topp:STATE_NAME</ogc:PropertyName>
                        <ogc:Literal>California</ogc:Literal>
                      </ogc:PropertyIsEqualTo>
                    </ogc:Filter>
                  </sld:FeatureTypeConstraint>
                </sld:LayerFeatureConstraints>
                <sld:UserStyle>
                  <sld:Name>topp:states</sld:Name>
                  <sld:Title>topp:states</sld:Title>
                  <sld:FeatureTypeStyle>
                    <sld:Rule>
                      <sld:PolygonSymbolizer>
                        <sld:Geometry>
                          <ogc:PropertyName>topp:the_geom</ogc:PropertyName>
                        </sld:Geometry>
                        <sld:Stroke>
                          <sld:CssParameter name="stroke">#0000FF</sld:CssParameter>
                          <sld:CssParameter name="stroke-width">2</sld:CssParameter>
                        </sld:Stroke>
                      </sld:PolygonSymbolizer>
                    </sld:Rule>
                  </sld:FeatureTypeStyle>
                </sld:UserStyle>
              </sld:UserLayer>
            </sld:StyledLayerDescriptor>
          </SLD>
          </Style>
        </StyleList>
        </Layer>
    </LayerList>
</ViewContext>
```

Figure 125 "ViewContext" document: ListReport_WMC.xml

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/xml" prefix="x"%>
<%@taglib prefix="context" tagdir="/WEB-INF/tags/context" %>
<html>
  <head><title>ViewContext debug file</title></head>
  <body>
    <context:ViewContext
        url="http://127.0.0.1:8084/deegreewms/debug/list_report/ListReport_WMC.xml">
      <context:GetFeature layerName="States" maxFeatures="100" viewContext="${viewContext}">
        <h3>WFS GetFeature request</h3>
```

```
            <c:out value="${features.request}" escapeXml="true"/>

            <h3>WFS GetFeature response</h3>
            <c:out value="${features.response}" escapeXml="true"/>

            <h3>GetFeature response XSL transformation</h3>
            <c:import var="xsl" url="/debug/list_report/List.xsl"/>
            <x:transform doc="${response}" xslt="${xsl}"/>
          </context:GetFeature>
      </context:ViewContext>
   </body>
</html>
```

Figure 126 Jsp file: List.jsp

The "LayerFeatureConstraints" section contains the information which features are available. With this information the WFS "GetFeatureRequest" can be created. The service URL of this scenario is "http://127.0.0.1:8084/geoserver/wfs".

```
<?xml version="1.0" ?>
<GetFeature xmlns="http://www.opengis.net/wfs" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:gml="http://www.opengis.net/gml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
version="1.0.0" service="WFS" output="GML2" maxFeatures="100">
   <Query typeName="topp:states">
      <ogc:Filter>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>topp:STATE_NAME</ogc:PropertyName>
          <ogc:Literal>California</ogc:Literal>
        </ogc:PropertyIsEqualTo>
      </ogc:Filter>
   </Query>
</GetFeature>
```

Figure 127 Browser output: List.jsp GetFeature request

The request returns the following response not including the detailed geometry description, which is skipped.

WFS GetFeature response

```
<?xml version="1.0" encoding="UTF-8"?><wfs:FeatureCollection
xmlns:wfs="http://www.opengis.net/wfs" xmlns:topp="http://www.openplans.org/topp"
xmlns:gml="http://www.opengis.net/gml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openplans.org/topp
http://127.0.0.1:8084/geoserver/wfs/DescribeFeatureType?typeName=topp:states
http://www.opengis.net/wfs http://127.0.0.1:8084/geoserver/data/capabilities/wfs/1.0.0/WFS-
basic.xsd"><gml:boundedBy><gml:Box
srsName="http://www.opengis.net/gml/srs/epsg.xml#4326"><gml:coordinates
xmlns:gml="http://www.opengis.net/gml" decimal="." cs="," ts=" ">-124.391472,32.535725 -
114.124451,42.002346</gml:coordinates></gml:Box></gml:boundedBy><gml:featureMember><topp:s
tates fid="states.47"><topp:the_geom><gml:MultiPolygon
srsName="http://www.opengis.net/gml/srs/epsg.xml#4326"><gml:polygonMember><gml:Polygon>
...The geometry …
</gml:MultiPolygon></topp:the_geom><topp:STATE_NAME>California</topp:STATE_NAME><top
```

p:STATE_FIPS>06</topp:STATE_FIPS><topp:SUB_REGION>Pacific</topp:SUB_REGION><topp:
STATE_ABBR>CA</topp:STATE_ABBR><topp:LAND_KM>403970.143</topp:LAND_KM>
<topp:WATER_KM>20023.368</topp:WATER_KM><topp:PERSONS>2.9760021E7</topp:PERSON
S><topp:FAMILIES>7139394.0</topp:FAMILIES><topp:HOUSHOLD>1.0381206E7</topp:HOUSH
OLD><topp:MALE>1.4897627E7</topp:MALE><topp:FEMALE>1.4862394E7</topp:FEMALE><to
pp:WORKERS>1.1306576E7</topp:WORKERS><topp:DRVALONE>9982242.0</topp:DRVALONE
><topp:CARPOOL>2036025.0</topp:CARPOOL><topp:PUBTRANS>685797.0</topp:PUBTRANS>
<topp:EMPLOYED>1.3996309E7</topp:EMPLOYED><topp:UNEMPLOY>996502.0</topp:UNEMP
LOY><topp:SERVICE>3664771.0</topp:SERVICE><topp:MANUAL>1798201.0</topp:MANUAL>
<topp:P_MALE>0.501</topp:P_MALE><topp:P_FEMALE>0.499</topp:P_FEMALE><topp:SAMP_
POP>3792553.0</topp:SAMP_POP><topp:PFEMALE>50</topp:PFEMALE><topp:PMALE>50</top
p:PMALE></topp:states></gml:featureMember></wfs:FeatureCollection>

Figure 128 Browser output: List.jsp GetFeature response

The WFS response can now be transformed with the following XSL style sheet.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:gml="http://www.opengis.net/gml" xmlns:wfs="http://www.opengis.net/wfs"
xmlns:topp="http://www.openplans.org/topp">
  <xsl:template match="/">
    <h1>State: <xsl:value-of
        select="/wfs:FeatureCollection/gml:featureMember/topp:states/topp:STATE_NAME"/></h1>
    <xsl:apply-templates select="/wfs:FeatureCollection/gml:boundedBy"/>
  </xsl:template>
  <xsl:template match="gml:boundedBy">
    <table border="1">
      <tr><th colspan="2">SRS / Bounding box</th></tr>
      <tr><td> <xsl:value-of select="gml:Box/@srsName"/> /
          <xsl:value-of select="gml:Box/gml:coordinates"/>
        </td></tr>
    </table>
  </xsl:template>
</xsl:stylesheet>
```

Figure 129 XSL stylesheet: List.xsl

State: California

SRS / Bounding box
http://www.opengis.net/gml/srs/epsg.xml#4326 / -124.391472,32.535725 -114.124451,42.002346

Figure 130 Browser output: List.jsp XSL transformation

With an “Extensible Stylesheet” (XSL) the response of the “GetFeature” request is transformed to a web page. Naturally, it is possible to generate other data formats e.g. comma separated value (CSV) files in the same way.

5.3 Data manipulation and analysis functions

5.3.1 Query

For the “query” test the “RemoteWFS_States.xml” view context file (Appendix: ViewContext documents pp. 155) is used. The query capabilities are examined with the help of four online referenced SLD files in the “ViewContext” document. They are added with the “AddStyle” (p. 143) tag. The first test performs an attribute query (“States_Attribute_Query_SLD.xml”) for all states whose name is equal to “California”. The next one performs a spatial query (“States_Spatial_Query_SLD.xml”) for all states that are situated within a given box. The third and final test performs a combined query that displays all states whose name is either "California" or are situated within a given box (“States_Combined_Query_SLD.xml”).

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="context" tagdir="/WEB-INF/tags/context" %>
<html>
  <head><title>Query debug file</title></head>
  <body>
    <h2>Query debug file</h2>
    <context:ViewContext
          url="http://127.0.0.1:8084/deegreewms/debug/wmc/RemoteWFS_States.xml">
      <h3>Original map from remote WFS</h3>
      <context:GetMap viewContext="${viewContext}"/>

      <h3>Attribute query from remote WFS</h3>
      <context:AddStyle layerName="States" viewContext="${viewContext}">
        <Style xmlns="http://www.opengis.net/context" xmlns:sld="http://www.opengis.net/sld"
            xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ogc="http://www.opengis.net/ogc"
            xmlns:gml="http://www.opengis.net/gml" >
          <SLD>
            <Name>AttributeQuery</Name>
            <OnlineResource xlink:type="simple"
      xlink:href="http://127.0.0.1:8084/deegreewms/debug/query/States_Attribute_Query_SLD.xml"/>
          </SLD>
        </Style>
      </context:AddStyle>
      <context:GetMap viewContext="${viewContext}"/>

      <h3>Spatial query from remote WFS</h3>
      <context:AddStyle layerName="States" viewContext="${viewContext}">
        <Style xmlns="http://www.opengis.net/context" xmlns:sld="http://www.opengis.net/sld"
            xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ogc="http://www.opengis.net/ogc"
            xmlns:gml="http://www.opengis.net/gml" >
          <SLD>
            <Name>SpatialQuery</Name>
            <OnlineResource xlink:type="simple"
```

```
          xlink:href="http://127.0.0.1:8084/deegreewms/debug/query/States_Spatial_Query_SLD.xml"/>
              </SLD>
            </Style>
          </context:AddStyle>
          <context:GetMap viewContext="${viewContext}"/>
```

```
          <h3>Combined query from remote WFS</h3>
          <context:AddStyle layerName="States" viewContext="${viewContext}">
            <Style xmlns="http://www.opengis.net/context" xmlns:sld="http://www.opengis.net/sld"
                xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ogc="http://www.opengis.net/ogc"
                xmlns:gml="http://www.opengis.net/gml" >
              <SLD>
                <Name>CombinedQuery</Name>
                <OnlineResource xlink:type="simple"
        xlink:href="http://127.0.0.1:8084/deegreewms/debug/query/States_Combined_Query_SLD.xml"/>
              </SLD>
            </Style>
          </context:AddStyle>
          <context:GetMap viewContext="${viewContext}"/>
        </context:ViewContext>
```

```
    </body>
</html>
```

Figure 131 Jsp file: Query.jsp

```
See appendix
```

```
        <LayerFeatureConstraints>
          <FeatureTypeConstraint>
            <FeatureTypeName>topp:states</FeatureTypeName>
          </FeatureTypeConstraint>
        </LayerFeatureConstraints>
```

```
See appendix
```

Figure 132 "SLD" file: States_SLD.xml

The "States_SLD.xml" SLD file is an example with no "ogc:Filter" in the "LayerFeatureConstraints" element. Therefore, all features are returned.

```
Same as above
        <LayerFeatureConstraints>
          <FeatureTypeConstraint>
            <FeatureTypeName>topp:states</FeatureTypeName>
```

```
            <ogc:Filter>
              <ogc:PropertyIsEqualTo>
                <ogc:PropertyName>topp:STATE_NAME</ogc:PropertyName>
                <ogc:Literal>California</ogc:Literal>
              </ogc:PropertyIsEqualTo>
            </ogc:Filter>
```

```
          </FeatureTypeConstraint>
        </LayerFeatureConstraints>
```

```
Same as above
```

Figure 133 "SLD" file fragment: States_Attribute_Query_SLD.xml

The attributive query adds a comparison operator to the "LayerFeatureContraints" element. As result only features with the property name "topp:STATE_NAME" that equals "California" are returned.

```
Same as above
    <LayerFeatureConstraints>
      <FeatureTypeConstraint>
        <FeatureTypeName>topp:states</FeatureTypeName>
        <ogc:Filter>
          <ogc:Within>
            <ogc:PropertyName>topp:the_geom</ogc:PropertyName>
            <gml:Box srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
              <gml:coord>
                <gml:X>-100</gml:X>
                <gml:Y>30</gml:Y>
              </gml:coord>
              <gml:coord>
                <gml:X>-60</gml:X>
                <gml:Y>60</gml:Y>
              </gml:coord>
            </gml:Box>
          </ogc:Within>
        </ogc:Filter>
      </FeatureTypeConstraint>
    </LayerFeatureConstraints>
Same as above
```

Figure 134 "SLD" file fragement: States_Spatial_Query_SLD.xml

A spatial query is created much similar to an attributive query, but in this case a spatial operator is applied (p. 57). Only those states are returned whose geometry ("topp:the_geom" property) is situated inside the given box ("gml:Box").

```
Same as above
    <LayerFeatureConstraints>
      <FeatureTypeConstraint>
        <FeatureTypeName>topp:states</FeatureTypeName>
        <ogc:Filter>
          <ogc:Or>
            <ogc:PropertyIsEqualTo>
              <ogc:PropertyName>topp:STATE_NAME</ogc:PropertyName>
              <ogc:Literal>California</ogc:Literal>
            </ogc:PropertyIsEqualTo>
            <ogc:Within>
              <ogc:PropertyName>topp:the_geom</ogc:PropertyName>
```

```
                    <gml:Box srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
                        … The geometry as above …
                    </gml:Box>
                </ogc:Within>
            <ogc:Or>
        </ogc:Filter>
```

```
    </FeatureTypeConstraint>
</LayerFeatureConstraints>
```

Same as above

Figure 135 "SLD" file fragement: States_Combinded_Query_SLD.xml

Here the "ogc:Or" logical operator is added to the "FeatureTypeContraint" element. Inside the operator an attribute and a spatial query are combined. The result is a map in which the result of both queries is displayed.

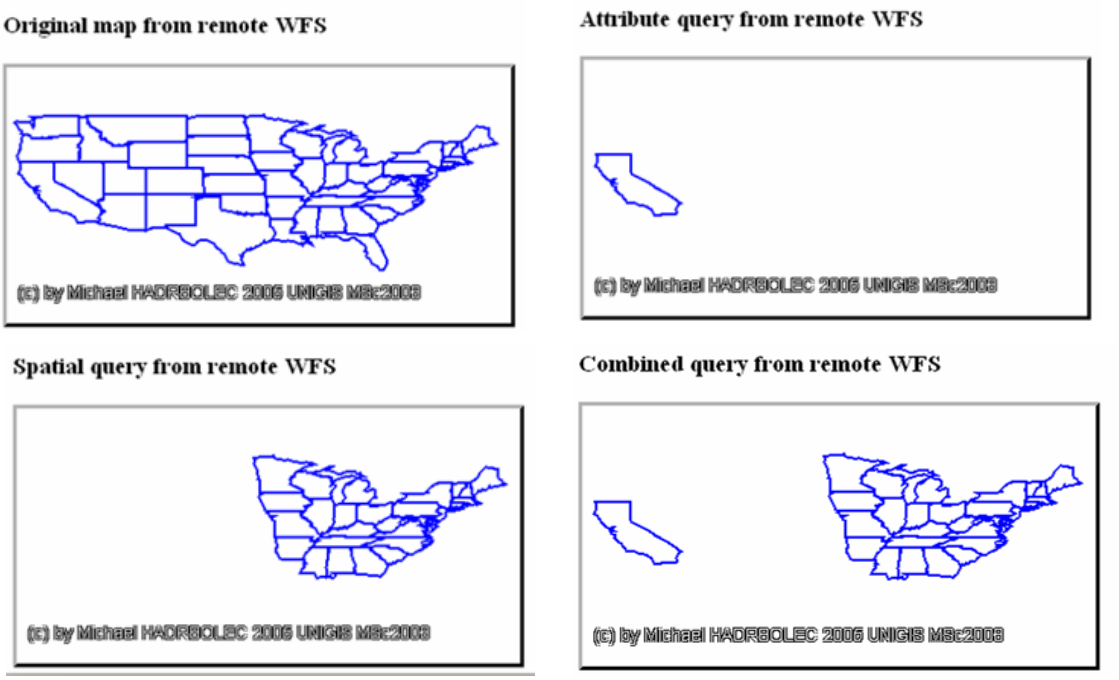


Figure 136 Browser output: Query.jsp

5.3.2 Generating features, views, and graphs

5.3.2.1 Buffer

Once more the "RemoteWFS_States.xml" ViewContext document is used (Appendix: ViewContext documents pp. 155). A buffer with the "ogc:DWithin" element is added by referencing an online accessible "States_Buffer.sld" SLD file to the "LayerFeatureConstrainsts" section. This is done with the "AddStyle" tag (Appendix: Tag files pp. 143). The buffer should be a circle with 7° angular distance units around a

centre point at -80° longitude and 40° latitude. The test failed due to implementation errors, a workaround is described later on.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="context" tagdir="/WEB-INF/tags/context" %>
<html>
  <head><title>Buffer debug file</title></head>
  <body>
    <h2>Buffer debug file</h2>
    <context:ViewContext
        url="http://127.0.0.1:8084/deegreewms/debug/wmc/RemoteWFS_States.xml">
    <h3>Original map from remote WFS</h3>
    <context:GetMap viewContext="${viewContext}"/>
    <h3>Buffer</h3>
    <context:AddStyle layerName="States" viewContext="${viewContext}">
      <Style xmlns="http://www.opengis.net/context" xmlns:sld="http://www.opengis.net/sld"
          xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ogc="http://www.opengis.net/ogc"
          xmlns:gml="http://www.opengis.net/gml" >
        <SLD>
          <Name>Buffer</Name>
          <OnlineResource xlink:type="simple"
            xlink:href="http://127.0.0.1:8084/deegreewms/debug/buffer/States_Buffer_SLD.xml"/>
        </SLD>
      </Style>
    </context:AddStyle>
    <context:GetMap viewContext="${viewContext}"/>

    <h3>Buffer workaround</h3>
    <context:AddStyle layerName="States" viewContext="${viewContext}">
      <Style xmlns="http://www.opengis.net/context" xmlns:sld="http://www.opengis.net/sld"
          xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ogc="http://www.opengis.net/ogc"
          xmlns:gml="http://www.opengis.net/gml" >
        <SLD>
          <Name>BufferWorkaround</Name>
          <OnlineResource xlink:type="simple"
xlink:href="http://127.0.0.1:8084/deegreewms/debug/buffer/States_Buffer_workaround_SLD.xml"/>
        </SLD>
      </Style>
    </context:AddStyle>
    <context:GetMap viewContext="${viewContext}"/>
    </context:ViewContext>
  </body>
</html>
```

Figure 137 Jsp file: Buffer.jsp

```
<?xml version="1.0" encoding="UTF-8"?>
<StyledLayerDescriptor version="1.0.0" xmlns="http://www.opengis.net/sld"
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" >
  <UserLayer>
    <RemoteOWS>
      <Service>WFS</Service>
      <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
          xlink:href="http://127.0.0.1:8084/geoserver/wfs"/>
    </RemoteOWS>
    <LayerFeatureConstraints>
      <FeatureTypeConstraint>
        <FeatureTypeName>topp:states</FeatureTypeName>
        <ogc:Filter>
          <ogc:DWithin>
            <ogc:PropertyName>topp:the_geom</ogc:PropertyName>
            <gml:Point srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
              <gml:coord>
                <gml:X>-80</gml:X>
                <gml:Y>40</gml:Y>
              </gml:coord>
            </gml:Point>
            <ogc:Distance
              units="http://www.uomdict.com/uom.html#decimal_deegrees">7</ogc:Distance>
          </ogc:DWithin>
        </ogc:Filter>
      </FeatureTypeConstraint>
    </LayerFeatureConstraints>
    <UserStyle>
      …The user defined style …
    </UserStyle>
  </UserLayer>
</StyledLayerDescriptor>
```

Figure 138 "SLD" file: States_Buffer_SLD.xml

```
<ServiceException
locator="org.vfny.geoserver.requests.readers.XmlRequestReader">org.xml.sax.SAXException:
Attempted to construct illegal filter: Got to the end state of an incomplete filter, current state is
distance</ServiceException></ServiceExceptionReport>
```

Figure 139 Thrown error message from the Geoserver

Unfortunately, this approach did not work. It seems to be an implementation error of either the Deegree or the Geoserver implementation (Figure 139).

```
<?xml version="1.0" encoding="UTF-8"?>
<StyledLayerDescriptor version="1.0.0" xmlns="http://www.opengis.net/sld"
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:xlink="http://www.w3.org/1999/xlink" >
  <UserLayer>
    <RemoteOWS>
      <Service>WFS</Service>
      <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
          xlink:href="http://127.0.0.1:8084/geoserver/wfs?"/>
    </RemoteOWS>
    <LayerFeatureConstraints>
      <FeatureTypeConstraint>
        <FeatureTypeName>topp:states</FeatureTypeName>
      </FeatureTypeConstraint>
    </LayerFeatureConstraints>
    <UserStyle>
      <Name>Buffer</Name>
      <Title>Buffer</Title>
      <FeatureTypeStyle>
        <Rule>
          <ogc:Filter>
            <ogc:DWithin>
              <ogc:PropertyName>topp:the_geom</ogc:PropertyName>
              <gml:Point srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
                <gml:coord>
                  <gml:X>-80</gml:X>
                  <gml:Y>40</gml:Y>
                </gml:coord>
              </gml:Point>
              <ogc:Distance
                  units="http://www.uomdict.com/uom.html#decimal_deegrees">7</ogc:Distance>
            </ogc:DWithin>
          </ogc:Filter>
          <PolygonSymbolizer>
              … The symbolizer …
          </PolygonSymbolizer>
        </Rule>
      </FeatureTypeStyle>
    </UserStyle>
  </UserLayer>
</StyledLayerDescriptor>
```

Figure 140 “SLD” file: States_Buffer_workaround_SLD.xml

The drawback of the workaround is that it is not possible to output the result as list.

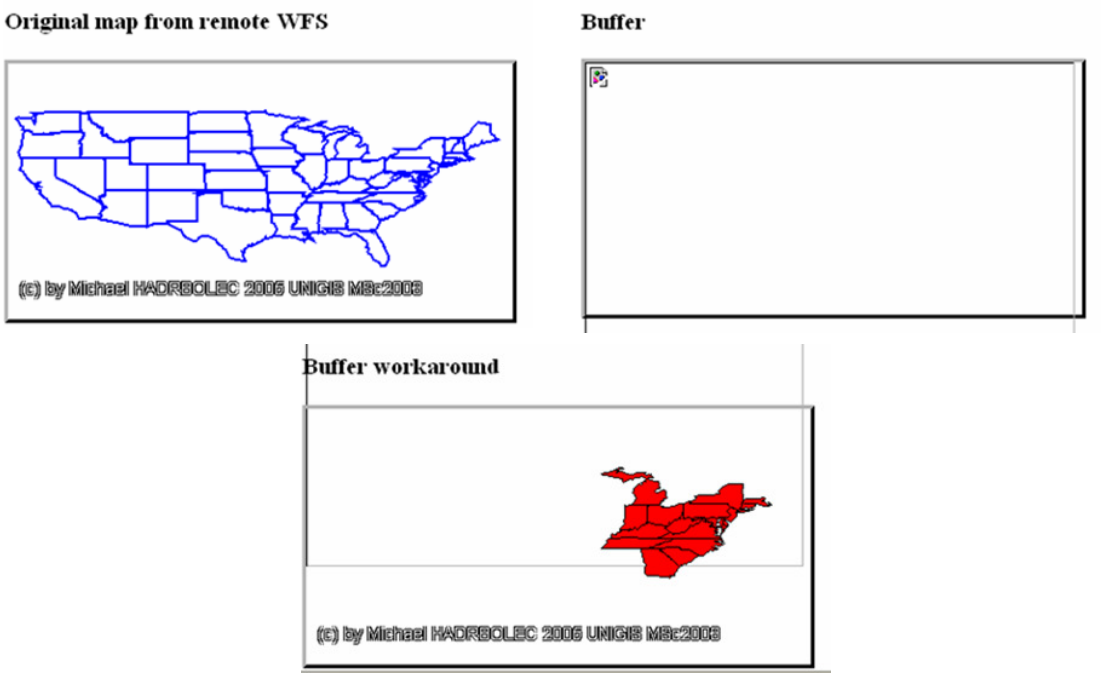


Figure 141 Browser output: Buffer.jsp

5.3.2.2 Generate graph

A graph can only be generated indirectly. As mentioned earlier the response of the WFS server is an XML file which can be transformed to a list (pp. 88) as well as to a graph using third-party software components.

5.3.3 Manipulating features

5.3.3.1 Classify attributes

Again, the "RemoteWFS_States.xml" view context document is used (Appendix: ViewContext documents pp. 155). The aim of this test case is to classify the "topp:States" layer according to the attribute field "Persons"which contains the number of inhabitants in the respective state. Less than 4.000.000 inhabitants should be visualised with red, between 4.000.0000 and 6.000.000 green and greater then 6.000.000 with blue colour. This is achieved by adding an online reference styled layer descriptor file with the "AddStyle" tag (pp. 143).

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="context" tagdir="/WEB-INF/tags/context" %>
<html>
   <head><title>Classify attribute debug file</title></head>
   <body>
      <h2>Classify attribute debug file</h2>
      <context:ViewContext
            url="http://127.0.0.1:8084/deegreewms/debug/wmc/RemoteWFS_States.xml">
```

```
        <h3>Original map from remote WFS</h3>
        <context:GetMap viewContext="${viewContext}"/>
        <h3>Classify attribute</h3>
        <context:AddStyle layerName="States" viewContext="${viewContext}">
          <Style xmlns="http://www.opengis.net/context" xmlns:sld="http://www.opengis.net/sld"
                 xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ogc="http://www.opengis.net/ogc"
                 xmlns:gml="http://www.opengis.net/gml" >
            <SLD>
              <Name>Classify</Name>
              <OnlineResource xlink:type="simple"

xlink:href="http://127.0.0.1:8084/deegreewms/debug/classify/States_Classify_Attribute_SLD.xml"/>
            </SLD>
          </Style>
        </context:AddStyle>
        <context:GetMap viewContext="${viewContext}"/>

        <h3>Classify attribute workaround</h3>
        <context:AddStyle layerName="States" viewContext="${viewContext}">
          <Style xmlns="http://www.opengis.net/context" xmlns:sld="http://www.opengis.net/sld"
                 xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ogc="http://www.opengis.net/ogc"
                 xmlns:gml="http://www.opengis.net/gml" >
            <SLD>
              <Name>ClassifyWorkaround</Name>
              <OnlineResource xlink:type="simple"
xlink:href="http://127.0.0.1:8084/deegreewms/debug/classify/States_Classify_Attribute_Workaround_SLD.xml"/>
            </SLD>
          </Style>
        </context:AddStyle>
        <context:GetMap viewContext="${viewContext}"/>
      </context:ViewContext>
    </body>
</html>
```

Figure 142 Jsp file: Classify.jsp

```
<?xml version="1.0" encoding="UTF-8"?>
<StyledLayerDescriptor version="1.0.0" xmlns="http://www.opengis.net/sld"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:xlink="http://www.w3.org/1999/xlink" >
  <UserLayer>
    <Name>ClassifyAttribute</Name>
    <RemoteOWS>
      <Service>WFS</Service>
      <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
          xlink:href="http://127.0.0.1:8084/geoserver/wfs"/>
    </RemoteOWS>
    <LayerFeatureConstraints>
      <FeatureTypeConstraint>
        <FeatureTypeName>topp:states</FeatureTypeName>
```

```
      </FeatureTypeConstraint>
    </LayerFeatureConstraints>
    <UserStyle>
      <Name>topp:states</Name>
      <Title>Classify</Title>
      <FeatureTypeStyle>
        <Rule>
          <Name>LessThan</Name>
          <ogc:Filter>
            <ogc:PropertyIsLessThan>
              <ogc:PropertyName>topp:PERSONS</ogc:PropertyName>
              <ogc:Literal>4000000</ogc:Literal>
            </ogc:PropertyIsLessThan>
          </ogc:Filter>
          <PolygonSymbolizer>
            <Geometry>
              <ogc:PropertyName>topp:the_geom</ogc:PropertyName>
            </Geometry>
            <Fill>
              <CssParameter name="fill">#FF0000</CssParameter>
              <CssParameter name="opacity">0.5</CssParameter>
            </Fill>
          </PolygonSymbolizer>
        </Rule>
        <Rule>
          <Name>Between</Name>
          <ogc:Filter>
            <ogc:PropertyIsBetween>
              <ogc:PropertyName>topp:PERSONS</ogc:PropertyName>
              <ogc:LowerBoundary>
                <ogc:Literal>4000000</ogc:Literal>
              </ogc:LowerBoundary>
              <ogc:UpperBoundary>
                <ogc:Literal>6000000</ogc:Literal>
              </ogc:UpperBoundary>
            </ogc:PropertyIsBetween>
          </ogc:Filter>
          <PolygonSymbolizer>
            <Geometry>
              <ogc:PropertyName>topp:the_geom</ogc:PropertyName>
            </Geometry>
            <Fill>
              <CssParameter name="fill">#00FF00</CssParameter>
              <CssParameter name="opacity">0.5</CssParameter>
            </Fill>
          </PolygonSymbolizer>
        </Rule>
```

```
            <Rule>
              <Name>GreaterThan</Name>
              <ogc:Filter>
                <ogc:PropertyIsGreaterThan>
                  <ogc:PropertyName>topp:PERSONS</ogc:PropertyName>
                  <ogc:Literal>6000000</ogc:Literal>
                </ogc:PropertyIsGreaterThan>
              </ogc:Filter>
              <PolygonSymbolizer>
                <Geometry>
                  <ogc:PropertyName>topp:the_geom</ogc:PropertyName>
                </Geometry>
                <Fill>
                  <CssParameter name="fill">#00FFFF</CssParameter>
                  <CssParameter name="opacity">0.5</CssParameter>
                </Fill>
              </PolygonSymbolizer>
            </Rule>
            <Rule>
              <PolygonSymbolizer>
                <Geometry>
                  <ogc:PropertyName>topp:the_geom</ogc:PropertyName>
                </Geometry>
                <Stroke>
                  <CssParameter name="color">#000000</CssParameter>
                  <CssParameter name="size">1</CssParameter>
                </Stroke>
              </PolygonSymbolizer>
            </Rule>
          </FeatureTypeStyle>
        </UserStyle>
      </UserLayer>
</StyledLayerDescriptor>
```

Figure 143 “SLD” file: States_Classify_Attribute_ SLD.xml

```
Error evaluating filter: org.deegree.services.wfs.filterencoding.FilterEvaluationException:
PropertyIsBetweenOperation can only be applied to numerical expressions
```

Figure 144 “SLD” file: States_Classify_Attribute_ SLD.xml

Unfortunately, this approach did not work. It seems once more to be a further implementation error as can be seen in Figure 144 above. As workaround the author registered the “States” layer as local “Web Feature Service” inside the Deegree WMS server (Integrated WMS) to successfully master the problem.

```
<?xml version="1.0" encoding="UTF-8"?>
<StyledLayerDescriptor version="1.0.0" xmlns="http://www.opengis.net/sld"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:xlink="http://www.w3.org/1999/xlink" >
   <NamedLayer>
     <Name>msc:states</Name>
     <UserStyle>
       <Name>Classify</Name>
       <Title>Classify</Title>
     <UserStyle>
        … Same as above ….
     </UserStyle>
   </NamedLayer>
</StyledLayerDescriptor>
```

Figure 145 "SLD" file: States_Classify_Attribute_Workaround_SLD.xml

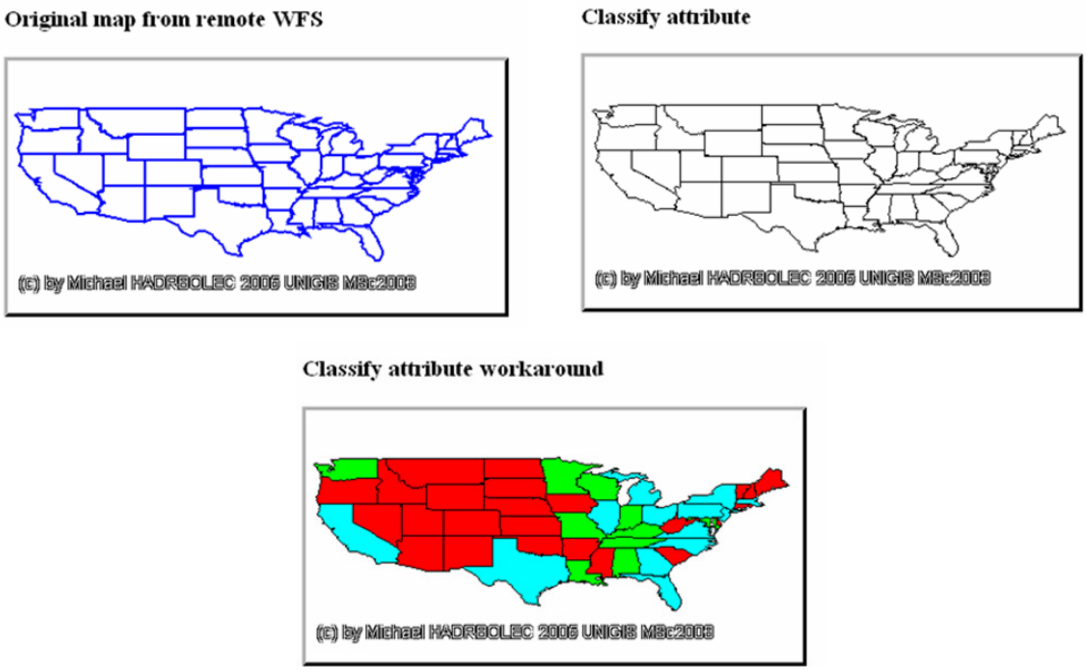


Figure 146 Browser output: Classify.jsp

The workaround returned the expected result.

5.3.3.2 Clip

See "Query" test case on page 92 for further details.

5.3.3.3 Scale change

5.3.3.3.1 Scale factor (zoom)

In this test scenario the "WMS_cite_WMC.xml" (Appendix: ViewContext documents pp. 155) view context document is used again. The first step is to zoom in with the factor 0.5 and afterward to zoom out with the factor 2. The result is a map in the original map scale. The centre of the displayed map remains unchanged. Therefore, the "Scale" tag (p. 147) is used.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="context" tagdir="/WEB-INF/tags/context" %>
<html>
  <head><title>Scale factor debug file</title></head>
  <body>
    <h2>Scale factor</h2>
    <context:ViewContext
      url="htpp://127.0.0.1:8084/deegreewms/debug/wmc/WMS_cite_WMC.xml">
      <h3>Original map</h3>
      <context:GetMap viewContext="${viewContext}"/>

      <h3>Scale factor 0.5 (zoomin factor 2)</h3>
      <context:Scale factor="0.5"  viewContext="${viewContext}"/>
      <context:GetMap viewContext="${viewContext}"/>

      <h3>Scale factor 2 (zoomout factor 2 = original map)</h3>
      <context:Scale factor="2"  viewContext="${viewContext}"/>
      <context:GetMap viewContext="${viewContext}"/>
    </context:ViewContext>
  </body>
</html>
```

Figure 147 Jsp file: ScaleFactorCenter.jsp

5.3.3.3.2 Scale factor centre

Once more the "WMS_cite_WMC.xml" view context document is used (Appendix: ViewContext documents pp. 155). In this test case the function zoom in with the factor 0.5 is invoked. The centre is moved to (400, 200). The next step is to zoom out again with the factor 2 and to move the centre to (0, 0). The final result will be the same as the original map. Therefore, the "Scale" tag (p. 147) is used.

```
<%@page contentType="text/html”  pageEncoding="UTF-8"%>
<%@taglib prefix="context" tagdir="/WEB-INF/tags/context" %>
<html>
   <head><title>Scale factor center debug file</title></head>
   <body>
      <h2>Scale factor</h2>
      <context:ViewContext
          url=”htpp://127.0.0.1:8084/deegreewms/debug/wmc/WMS_cite_WMC.xml“>
         <h3>Original map</h3>
          <context:GetMap viewContext="${viewContext}"/>

         <h3>Scale factor 0.5 right bottom border (x=400px, y=200px)</h3>
          <context:Scale factor="0.5" x="400" y="200" viewContext="${viewContext}"/>
         <context:GetMap viewContext="${viewContext}"/>

         <h3>Scale factor 2 (zoomout factor 2 x=0, y=0 = original map)</h3>
          <context:Scale factor="2" x="0" y="0"  viewContext="${viewContext}"/>
         <context:GetMap viewContext="${viewContext}"/>
      </context:ViewContext>
   </body>
</html>
```

Figure 148 Jsp file: ScaleFactorCenter

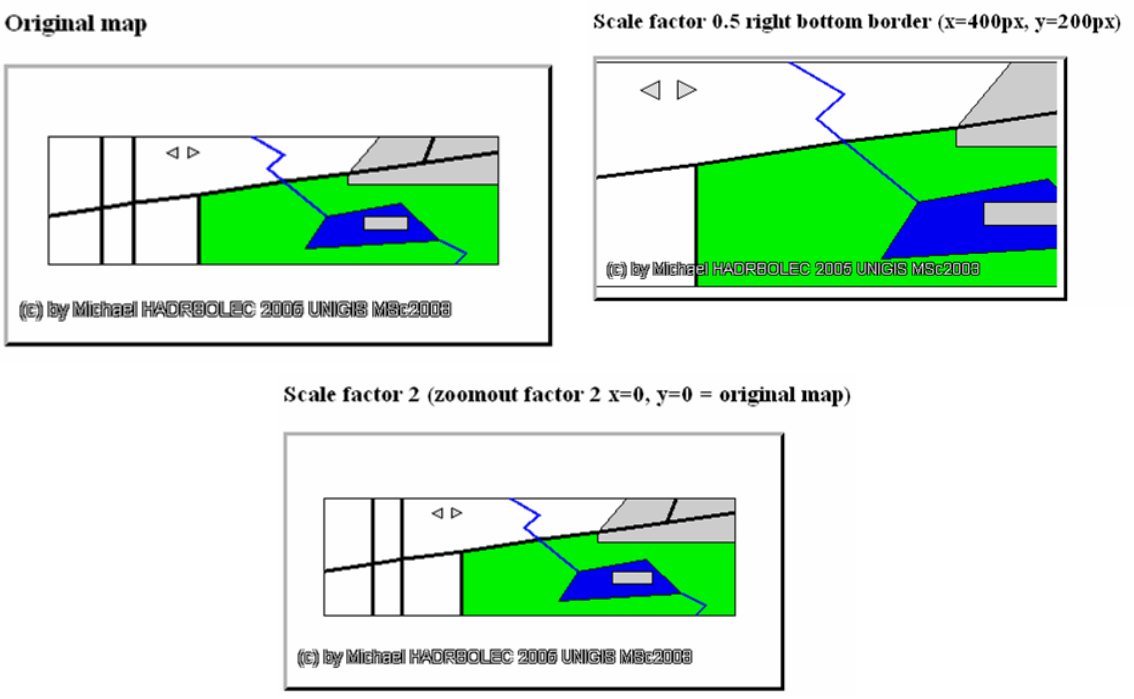


Figure 149 Browser output: ScaleFactorCenter.jsp

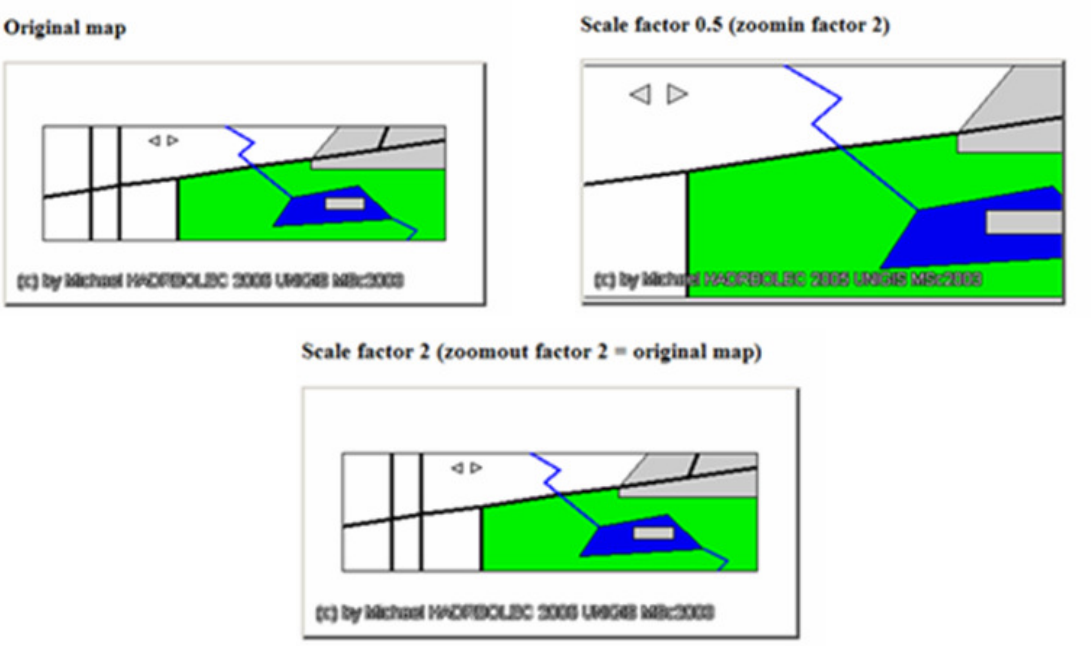


Figure 150 Browser output: ScaleFactorCenter.jsp

5.3.3.3.3 Scale to bounding box

The "WMS_cite_WMC.xml" ViewContext document is used for the next scenario (p. 157). This test emulates the "zoom to rectangle" function. Beginning with an original map "zoom in" starts with a given bounding box (minx="100" miny="50" maxx="300" maxy="150"). Subsequently, the original extent is re-established. For testing once more the "Scale" tag (p. 147) is used.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="context" tagdir="/WEB-INF/tags/context" %>
<html>
  <head><title>Scale factor bounding box debug file</title></head>
  <body>
    <h2>Scale bounding box</h2>
    <context:ViewContext
      url="htpp://127.0.0.1:8084/deegreewms/debug/wmc/WMS_cite_WMC.xml">
      <h3>Original map</h3>
      <context:GetMap viewContext="${viewContext}"/>

      <h3>Scale bounding box minx="100" miny="50" maxx="300" maxy="150"</h3>
      <context:Scale minx="100" miny="50" maxx="300" maxy="150"
        viewContext="${viewContext}"/>
      <context:GetMap viewContext="${viewContext}"/>

      <h3>Scale bounding box minx="-200" miny="-100" maxx="600" maxy="300"
        (=original map)</h3>
      <context:Scale minx="-200" miny="-100" maxx="600" maxy="300"
        viewContext="${viewContext}"/>
      <context:GetMap viewContext="${viewContext}"/>
```

```
    </context:ViewContext>
  </body>
</html>
```

Figure 151 Jsp file: ScaleBoundingBox.jsp

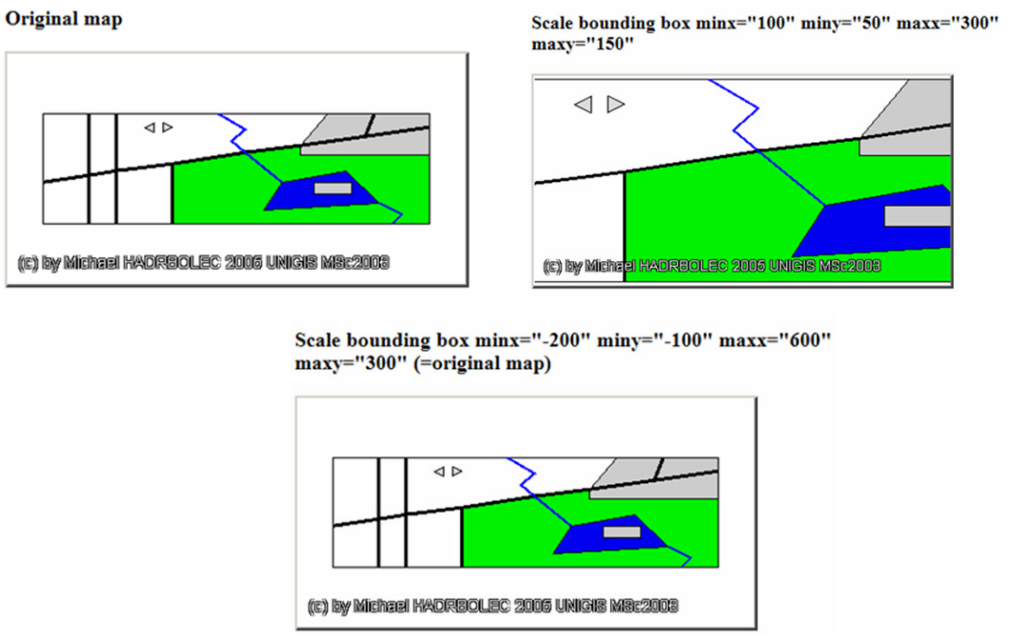


Figure 152 Browser output: ScaleBoundingBox.jsp

5.3.3.4 Projection change

A test case is not possible due to the fact that the specification is incorrect (p. 35).

5.3.4 Address locations

5.3.4.1 Address geocode (search)

This test case uses the "RemoteWFS_States.xml" view context file (pp. 155). All states that are named like "Cali*" will be geocoded. The first step is to perform a "GetFeature" request to the WFS server.

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
xmlns:topp="http://www.openplans.org/topp" xmlns:wfs="http://www.opengis.net/wfs"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs
   http://schemas.opengis.net/wfs/1.0.0/WFS-basic.xsd">
   <wfs:Query typeName="topp:states">
      <ogc:Filter>
         <ogc:PropertyIsLike wildCard="*" singleChar="#" escapeChar="!">
            <ogc:PropertyName>topp:STATE_NAME</ogc:PropertyName>
            <ogc:Literal>Cali*</ogc:Literal>
         </ogc:PropertyIsLike>
```

```
      </ogc:Filter>
   </wfs:Query>
</wfs:GetFeature>
```

Figure 153 WFS GetFeature request

The Web Feature Server (WFS) returns the following response.

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:FeatureCollection xmlns:wfs="http://www.opengis.net/wfs"
xmlns:topp="http://www.openplans.org/topp" xmlns:gml="http://www.opengis.net/gml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openplans.org/topp
http://127.0.0.1:8084/geoserver/wfs/DescribeFeatureType?typeName=topp:states
http://www.opengis.net/wfs http://127.0.0.1:8084/geoserver/data/capabilities/wfs/1.0.0/WFS-basic.xsd">
   <gml:boundedBy>
      <gml:Box srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
         <gml:coordinates xmlns:gml="http://www.opengis.net/gml" decimal="." cs="," ts=" ">
            -124.391472,32.535725 -114.124451,42.002346</gml:coordinates>
      </gml:Box>
   </gml:boundedBy>
   <gml:featureMember>
      <topp:states fid="states.47">
         <topp:the_geom>
                  … The geometry …
         </topp:the_geom>
         <topp:STATE_NAME>California</topp:STATE_NAME>
         <topp:STATE_FIPS>06</topp:STATE_FIPS>
                  … And so on …
      </topp:states>
   </gml:featureMember>
</wfs:FeatureCollection>
```

Figure 154 WFS GetFeatureRequest response

The next step is to add a new Layer ("Geocode") which diplays and highlights only the matching features. Therefore, inside the "LayerFeatureConstraints" section an "ogc:Filter" comparison filter is added. As final step a buffer of 1° is added to the - from the WFS response returned - bounding box (minx=-125.391472, miny=31.535725, maxx=-113.124451, maxy=43.002346). The view context is scaled with the "Scale" (p. 147) tag to this bounding box (zoom to bounding box, see p. 106). Finally, the map with the geocoded features is displayed.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="context" tagdir="/WEB-INF/tags/context" %>
<html>
   <head><title>Geocode debug file</title></head>
   <body>
      <h2>Geocode debug file</h2>
      <context:ViewContext
```

```
        url="http://127.0.0.1:8084/deegreewms/debug/wmc/RemoteWFS_States.xml">
        <h3>Original map from remote WFS</h3>
        <context:GetMap viewContext="${viewContext}"/>

        <h3>Geocoded map</h3>
        <context:AddLayer viewContext="${viewContext}">
          <Layer queryable="true" hidden="false" xmlns="http://www.opengis.net/context"
              xmlns:sld="http://www.opengis.net/sld" xmlns:xlink="http://www.w3.org/1999/xlink"
              xmlns:ogc="http://www.opengis.net/ogc">
            <Server service="OGC:WMS" version="1.1.1">
              <OnlineResource xlink:type="simple"
                  xlink:href="http://127.0.0.1:8084/deegreewms/wms?"/>
            </Server>
            <Name>Geocode</Name>
            <Title>States</Title>
            <SRS>EPSG:4326</SRS>
            <FormatList>
              <Format current="true">image/gif</Format>
            </FormatList>
            <StyleList>
              <Style current="true">
                <SLD>
                 <Name>default</Name>
                 <sld:StyledLayerDescriptor version="1.0.0">
                <sld:UserLayer>
                  <sld:RemoteOWS>
                    <sld:Service>WFS</sld:Service>
                    <sld:OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
                      xlink:type="simple" xlink:href="http://127.0.0.1:8084/geoserver/wfs?"/>
                  </sld:RemoteOWS>
                  <sld:LayerFeatureConstraints>
                    <sld:FeatureTypeConstraint>
                      <sld:FeatureTypeName>topp:states</sld:FeatureTypeName>
                      <ogc:Filter>
                        <ogc:PropertyIsLike wildCard="*" singleChar="#" escape="?" >
                          <ogc:PropertyName>topp:STATE_NAME</ogc:PropertyName>
                          <ogc:Literal>Cali*</ogc:Literal>
                        </ogc:PropertyIsLike>
                      </ogc:Filter>
                    </sld:FeatureTypeConstraint>
                  </sld:LayerFeatureConstraints>
                  <sld:UserStyle>
                    <sld:Name>Geocode</sld:Name>
                    <sld:Title>Geocode</sld:Title>
                    <sld:FeatureTypeStyle>
                      <sld:Rule>
                        <sld:PolygonSymbolizer>
                          <sld:Geometry>
```

```
                                <ogc:PropertyName>topp:the_geom</ogc:PropertyName>
                            </sld:Geometry>
                            <sld:Fill>
                                <sld:CssParameter name="fill">#FF0000</sld:CssParameter>
                                <sld:CssParameter name="opacity">0.5</sld:CssParameter>
                            </sld:Fill>
                        </sld:PolygonSymbolizer>
                    </sld:Rule>
                </sld:FeatureTypeStyle>
            </sld:UserStyle>
        </sld:UserLayer>
    </sld:StyledLayerDescriptor>
    </SLD>
    </Style>
  </StyleList>
 </Layer>
</context:AddLayer>
<context:GetMap viewContext="${viewContext}"/>
```

```
<h3>Scale to bounding box (=final map)</h3>
<context:Scale minx="-125.391472" miny="31.535725" maxx="-113.124451"
    maxy="43.002346" units="mapunits" viewContext="${viewContext}"/>
<context:GetMap viewContext="${viewContext}"/>
```

```
  </context:ViewContext>
 </body>
</html>
```

Figure 155 JspFile: AddressGeocode.jsp

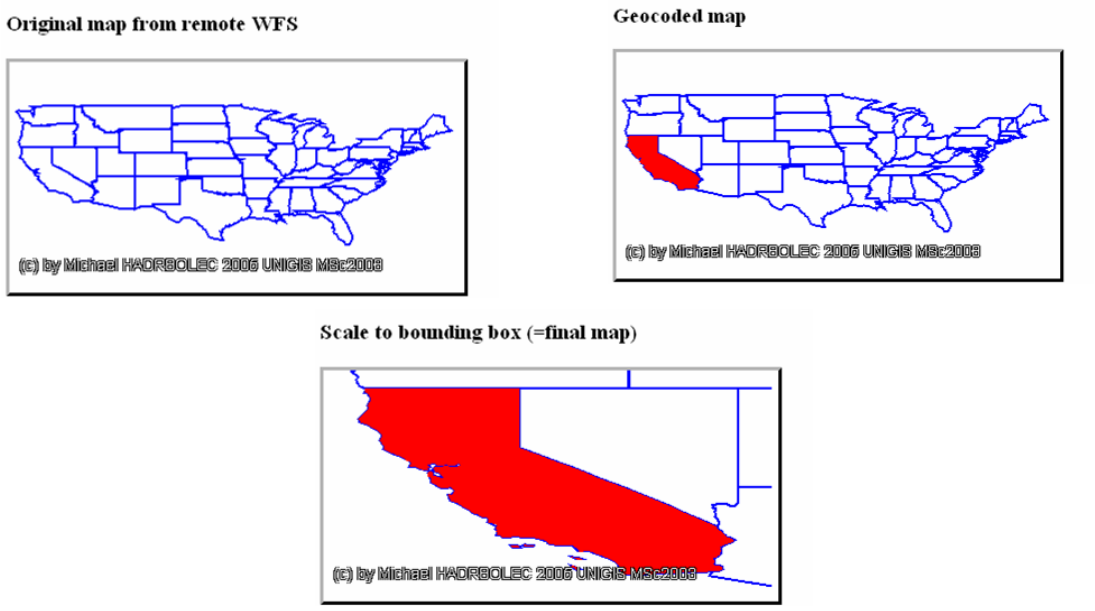


Figure 156 Browser output: AddressGeocode.jsp

5.3.5 Measurement

5.3.5.1.1 Measure length

This test case uses the "WMS_cite_WMC.xml"view context document (pp. 157). The measure function is checked with the horizontal distance of 400 pixel and the vertical distance of 200 pixels and the diagonal of the visible map extent. Therefore, the "Measure" (p. 149) tag is used.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="context" tagdir="/WEB-INF/tags/context" %>
<html>
  <head><title>Measure debug file</title></head>
  <body>
    <context:ViewContext
        url="http://localhost:8084/deegreewms/debug/wmc/WMS_cite_WMC.xml">
      <context:Measure dx="400" dy="0" viewContext="${viewContext}">
        Distance dx=400px, dy=0px: ${distance}<br/>
      </context:Measure>

      <context:Measure dx="0" dy="200" viewContext="${viewContext}">
        Distance dx=0px, dy=200px: ${distance}<br/>
      </context:Measure>

      <context:Measure dx="400" dy="200" viewContext="${viewContext}">
        Distance dx=400px, dy=200px: ${distance}<br/>
      </context:Measure>
    </context:ViewContext>
  </body>
</html>
```

Figure 157 Jsp file: Measure.jsp

```
<Window width="400" height="200"/>
<BoundingBox SRS="EPSG:4326" minx="-0.005" miny="-0.005" maxx="0.005" maxy="0.005"/>
```

Figure 158 Referenced "ViewContext" document fragment

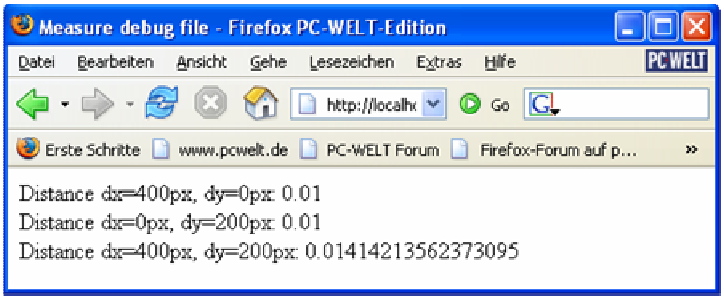


Figure 159 Browser output: Measure.jsp

As shown in Figure 159 the width and height in map units equals the width (0.005- (-0.005) =0.01) and height (0.005- (-0.005) =0.01) of the bounding box in map units of the referenced view context file (p. 157). The diagonal of the displayed extent returns also the correct value ($\sqrt{0.01^2 + 0.01^2} = 0.014142$).

5.3.6 Spatial Analysis

5.3.6.1 Graphic over plot

In this test case the "WMS_cite_WMC.xml" view context document (p. 157) is used. All layers that are visible (hidden attribute="false") are displayed and graphically overlaid. This function is carried out with the "GetMap" tag (p. 150).

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="context" tagdir="/WEB-INF/tags/context" %>
<html>
  <head><title>ViewContext debug file</title></head>
  <body>
    <context:ViewContext
      url="htpp://127.0.0.1:8084/deegreewms/debug/wmc/WMS_cite_WMC.xml">
      <context:GetMap context="${viewContext}"/>
    </context:ViewContext>
  </body>
</html>
```

Figure 160 Jsp file: GraphicalOverplot.jsp

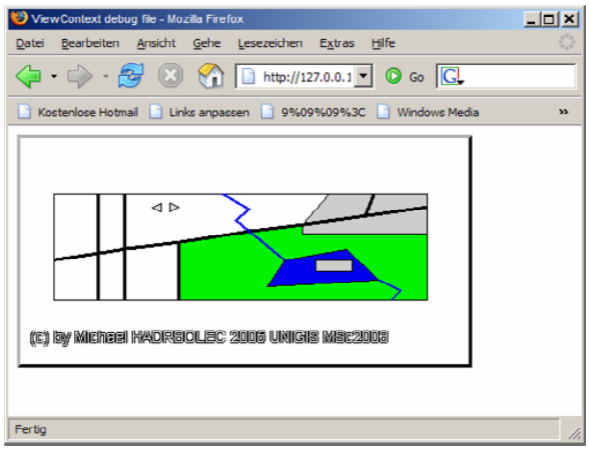


Figure 161 Browser output: GraphicOverplot.jsp

```
<div>
<img style="position:absolute;top:0px;left:0px"
src="http://127.0.0.1:8084/deegreewms/wms?SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&BBox=-0.0050%2C-0.0050%2C0.0050%2C0.0050&SRS=EPSG%3A4326&FORMAT=image%2Fgif&width=400&height=200&TRANSPARENT=true&LAYERS=cite%3ABasicPolygons&STYLES=default%3Acite%3ABasicPolygons&" width="400" width="200"/>

<img style="position:absolute;top:0px;left:0px"
src="http://127.0.0.1:8084/deegreewms/wms?SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&BBox=-0.0050%2C-0.0050%2C0.0050%2C0.0050&SRS=EPSG%3A4326&FORMAT=image%2Fgif&width=400&height=200&TRANSPARENT=true&LAYERS=cite%3ABridges&STYLES=default%3Acite%3ABuildings&" width="400" width="200"/>

… and all the other visible layers …
</div>
```

Figure 162 HTML fragment: GraphicOverplot.jsp

For each “Layer” with the hidden attribute set to “false” a WMS “GetMap” request is generated. The first layer is on bottom position, the last one is on top position.

6 Evaluation and further outlook

The final chapter first evaluates the results of the previous theoretical and practical review chapters. It continues afterwards with an outlook on further developments.

6.1 Evaluation

The evaluation is divided into the following parts:

- **Overview**
- **Use cases**
- **Further outlook**

6.1.1 Overview

The overview provides a summary of the functions identified as required in a list. It shows which functions are covered by the specification (Theoretical review), what are the technical prerequisites (Requirements) and whether they can be performed with the OGC WMS and WFS reference implementations (Practical review).

Function	Require-ments	Theoretical review	Practical review
Basic system capabilities			
Data input			
Load WMC file	WMC file	Yes	Success
Meta-information (WMC file)	WMC file	Partly	*1
Exchange WMC file	WMC file	Yes	Success *2
Storing maintaining and outputting data			
Serialize WMC file for later use (Store)	WMC file	Yes	Success
Edit and display (on output)			
Visibility			
Display	1, 2, 3 (a,b)	Yes	Success
Hide	1, 2, 3 (a,b)	Yes	Success
Add remove layer			
Add layer			
Add predefined layer	1, 2, 3 (a)	Yes	Success
Add user defined layer	3 (b)	Yes	Success
Remove layer	1, 2, 3 (a,b)	Yes	Success
Remove layer	1, 2, 3 (a,b)	Yes	Success

Function	Require-ments	Theoretical review	Practical review
Order	1, 2, 3 (a,b)	Yes	Success
Symbolize			
Predefined	1	Yes	Success
User defined			
Online StyledLayerDescriptor file	3 (a,b)	Yes	Success
Inline StyledLayerDescriptor	3 (a,b)	Yes	Success
Inline FeatureTypeDescriptor	3 (a,b)	Yes	Success
Annotate highlight	3 (a,b)	Yes	Success
Plot	1, 2, 3 (a,b)	Yes	Partly *3
Browse			
Move			
Move dx, dy	1, 2, 3 (a,b)	Yes	Success
Move from point to point	1, 2, 3 (a,b)	Yes	Success
Move to point (centre at)	1, 2, 3 (a,b)	Yes	Success
Identify	2	Yes	Success
Suppress	3 (a,b)	Yes	Partly *4
Create list (report)	3 (a,b)	Yes	Success
Data manipulation and analysis functions			
Query			
Spatial query	3 (a,b)	Yes	Success
Attribute query	3 (a,b)	Yes	Success
Combined query	3 (a,b)	Yes	Success
Generating features, views, and graphs			
Generate buffer	3 (a,b)	Yes	Partly *4
Generate graph	3 (a,b)	Yes	Skipped *3
Manipulate features			
Classify attributes	3 (a,b)	Yes	Partly *4
Clip	3 (a,b)	Yes	See query
Scale change			
Scale factor	1, 2, 3 (a,b)	Yes	Success
Scale centre factor	1, 2, 3 (a,b)	Yes	Success
Scale to bounding box	1, 2, 3 (a,b)	Yes	Success
Scale to specified scale	1, 2, 3 (a,b)	Yes	Success
Projection change	1, 2, 3 (a,b)	Partly *5	Skipped *5
Address location			

Function	Require-ments	Theoretical review	Practical review
Address geocode (search)	3 (a,b)	Yes	Success
Measurement			
Measure length	1, 2, 3 (a,b)	Yes	Success
Spatial analysis			
Graphical over plot	1, 2, 3 (a, b)	Yes	Success

Table 8 Summary of GIS function review

1 WMS basic profile
2 GetFeatureInfo enabled WMS server
3 SLD enabled WMS
a Integrated Server (datasource has to be in addition available as WFS / WCS)
b Component Server (datasource is as WFS/ WCS available)

*1 only reviewed theroretically
*2 exchange is possible (serialize / deserialize)
*3 in principle possible
*4 reference implementation error
*5 error in WMC specification (in principle possible)

As can be seen in Table 8, principally all necessary GIS functions are available. The author discovered only minor problems during the practical review part of his thesis. Implementation errors turned up inside the reference implementations (Deegree WMS and GeoServer WFS) and have nothing to do with the specification itself.

6.1.2 Use cases

As next step the results are discussed for the three reviewed use cases:

- "Browse"
- "Report" and
- "Explore" in further details.

6.1.2.1 Use case: "Browse"

The needed GIS functions of the "Browse" use case (p. 21) are completely covered by the "Web Map Context Documents" (WMC) specification. The technical requirement for OGC OWS services is a "GetFeatureInfo" enabled "Web Map Server" (WMS).

6.1.2.1.1 "Browse" use case gaps

The author identified no gaps.

6.1.2.2 Use case: “Report”

Beside the graphical display of a map, the “report” use case (p. 23) needs the possibility to render the features into lists, reports and graphs. Additionally, in some cases it is necessary to process the derived information further with other software products (e.g. spreadsheet- and text-processing software, etc.). Therefore, it is necessary to access data not only as map representation but also as datasets by using the “Web Feature Service” (WFS) for vector datasets and the “Web Coverage Service” (WCS) for raster datasets. The returned XML files can be transformed with the “Extensible Stylesheet Language” (XSL) into any format required by the user (p. 55; p. 59). The meta-information elements provide enough possibilities to comment the respective “ViewContext”. Teamwork is possible, too. The files can be either traditionally exchanged and / or synchronised simultaneously between the respective users (p. 38). The technical requirements are a "GetFeatureInfo" and “Styled Layer Descriptor” (SLD) enabled WMS server. The datasets which have to be provided as lists, graphs and / or reports, must also be available as WFS / WCS service.

6.1.2.2.1 “Report” use case gaps

6.1.2.2.1.1 Meta-information

Additionally to the provided meta-information elements (pp. 31), it is **required** to **know** the **name, geometry and data type** of the available **layers attributes** (geometry [vector: point polyline, polygon … / raster], date, character, double integer, etc.). Only with this kind of meta-information advanced classification, symbolisation and querying functions can be carried out. The **meta-information** of a WMC document differs from WMS, WFS and WCS services. A WMC document uses **different XML elements and namespaces** even **for the same elements** *e.g.”wms:Abstract” WMS , “context:Abstract”*. Meta-information elements have to be harmonised and a base profile is needed. This would foster reuse and simplifies the implementation for software providers.

6.1.2.2.1.2 HTTP POST Request for WMS undefined

It is possible to provide a user-defined symbolisation inline in a HTTP GET request as “key value” pair with the “SLD_BODY” element. The limiting factor is the browsers cache. In order to avoid this shortcoming it is necessary to define the HTTP POST request for the WMS 1.1.1 service (Symbolize p. 75).

6.1.2.2.1.3 Errors in OWS reference implementations

Due to implementation errors in either the Deegree (WMS) and / or the GeoServer (WFS) it was possible to perform suppress (p. 88), buffer (p. 95) and classify (p. 99) functions only with workarounds.

6.1.2.3 Use case: "Explore"

For the last evaluated "Explore" use case additionally to the functionality identified in the "Report" uses case, it is necessary to add and visualize layers (datasets) on the fly. This means it is required to be able to add a layer based on a WFS or WCS OWS service. Due to this fact the WMS service has to be able to visualize a remote WFS and / or WMC service according to the user's preferences (component server).

6.1.2.3.1 "Explore" use case gaps

In addition to all gaps identified in the "Report" use case the following gap was found:

6.1.2.3.1.1 Error in WMC XML schema

The WMC XML schema contains an error that allows to store only one "Spatial Reference System" (SRS) per layer. This avoids storing of all supported SRSs of a layer. The consequence is that it is not possible to reproject because it is unknown whether the layer supports the SRS or not (Layer-specific Meta-information pp. 35).

6.2 Further outlook

A **new** version (Version 1.3.0) of the **WMS service** has **recently** been **released** from the Open Geospatial Consortium (Beaujardiere, 2004). The new specification has become an **ISO standard** (ISO/DIS 19128 Geographic information – Web Map Server Interface). **Unfortunately,** this new version of the WMS service supports only the "Basic WMS" ("GetMap" and "GetCapabilities") and the "Queryable WMS" ("GetFeatureInfo"). This means that the new WMS 1.3.0 specification has **dropped** the **support for a SLD** enabled WMS server (Beaujardiere pp.26) which was optional in the WMS 1.1.1. The **consequence** is that it is **unclear how a WFS and / or WCS service can be visualised as map** in the future.

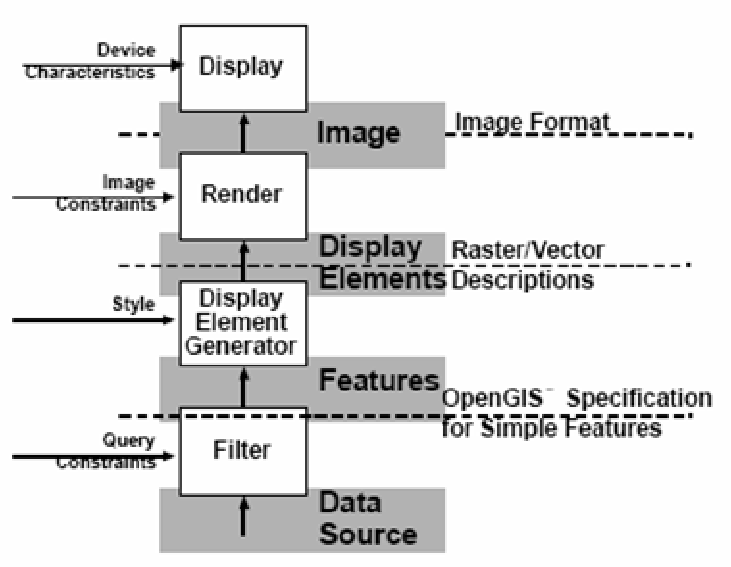


Figure 163 Portayal model (Buehler Figure 13 p. 23)

According to the OGC reference model, currently *"[…] no stand-alone conceptual model adopted by OGC for portrayal"exists* (Buehler et al. 2003, pp. 22). According to the author's opinion the OGC has to specify a new way how to render WFS and WCS. Otherwise, in the future complex user requirements like the one evaluated in this thesis are not feasible based on OGC standards and will therefore be implemented in a proprietary way.

7 Conclusions

At the present stage the OGC "**Web Map Context Documents" specification** version 1.0.0 **is able to fulfil** the **requirements** of the **evaluated** use cases of a groundwater expert:

- "**Browse**",
- "**Report**" and
- "**Explore**"

This implies that the **specification passed** the author's **reality check.** The only requirements are a "GetFeatureInfo" and a SLD enabled Web Mapping Service like the OGC WMS reference implementation of "Deegree". All layers which have to be displayed as lists, graphs or be further processed have to be additionally available as WFS or WCS services. Some minor obstacles had to be overcome due to errors in the "Deegree" and "GeoServer" implementations. Unfortunately, the **future** does not look so bright because of the **lacking support of SLD** in the **WMS 1.3.0** specification. There is currently no alternative OGC portrayal service for rendering WFS and WCS services available.

References

Beaujardiere (2002) OpenGIS Implementation Specification: Web Map Service, Version 1.1.1 [Internet] Open Geospatial Consortium. Available from: <http://portal.opengeospatial.org/files?artifact_id=5316> [Accessed November 8th, 2004]

Beaujardiere (2004) OpenGIS Implementation Specification: Web Map Service, Version 1.3.0 [Internet] Open Geospatial Consortium. Available from: <http://portal.opengeospatial.org/files?artifact_id=5316> [Accessed November 8th, 2004]

Beddoe, Cotton, Uleman, Johnson, Herring (1999) OpenGIS: . Simple Features Specification For SQL, Revision 1.1 [Internet] Open Geospatial Consortium <http://www.opengeospatial.org/docs/01-019.pdf> [Accessed November 8th, 2004]> [Accessed December 10th, 2004]

Buehler (2003) OpenGIS Reference Model, Version 0.1.2 [Internet] Open Geospatial Consortium <http://portal.opengis.org/files/?artifact_id=3836> [Accessed Jannuary 23th, 2005]

Cite Homepage (2004), Compliance Testing Initiative: Reference Implementations [Internet] Open Geospatial Consortium. <http://cite.occamlab.com/reference/index.html> [Accessed January 3rd, 2005]

Danko (Unknown) Perspectives in the development of ISO metadata standards [Internet] National Imagery and Mapping Agency,
Standards and Interoperability Division. Available from: <http://fgdc.er.usgs.gov/publications/documents/metadata/nimapaper.html> [Accessed January 12th, 2005]

Evans (2001) Discussion Paper: XML for Image and Map Annotations (XIMA)
Draft Candidate Interface Specification, Version 0.4 [Internet] Open Geospatial Consortium. <http://www.opengeospatial.org/docs/01-019.pdf> [Accessed November 8th, 2004]> [Accessed November 8th, 2004]

Fowler, Scott (2000) UML Distilled Second Edition: A brief guide to the standard modelling language. Adison-Wesley Longman, Inc., USA

Humblet (2003) OpenGIS Implementation Specification: Web Map Context Documents, Version 1.0.0 [Internet] Open Geospatial Consortium. Available from: <https://portal.opengeospatial.org/files/?artifact_id=3841> [Accessed November 8th, 2004]

Lalonde (2002) OpenGIS Implementation Specification: Styled layer Descriptor Implementation Specification, Version 1.0.0 [Internet] Open Geospatial Consortium. Available from: <http://portal.opengeospatial.org/files?artifact_id=1188> [Accessed November 8th, 2004]

Mc Govern, Tyag, Stevens, Mathew (2003), Java Web Services Architecture [Internet] Available from: <http://java.sun.com/developer/Books/j2ee/jwsa/JWSA_CH02.pdf> [Accessed December 12th, 2004]

Naccarato (2004) Template-Based Code Generation with Apache Velocity, Part 1 [Internet] Published on ONJava.com <http://www.onjava.com/pub/a/onjava/2004/05/05/cg-vel1.html> [Accessed Jannuary 23th, 2005]

Tomlinson (2003) Thinking About GIS: Geographic Information System Planning for Managers. Redlands California, ESRI Press.

Vretanos (2001) OpenGIS Implementation Specification: Filter Encoding Implementation Specification, Version 1.0.0 [Internet] Open Geospatial Consortium. Available from: <http http://www.opengeospatial.org/docs/02-059.pdf> [Accessed November 8th, 2004]

Bibliography

Alur, Crupi, Malks (2003): Core J2EE Patterns, Best Practics and Design Strategies. Palo Alto California USA, Sun Microsystems

Bernhard, Fitzke, Wager (2005) Geodateninfrastruktur, Grundlagen und Anwendungen. Heidelberg BRD, Wichmann

Bray, Paoli, Sperberg-McQueen, Maler, Yergeau (2004) Extensible Markup Language (XML) 1.0 (Third Edition) W3C Recommendation 04 February 2004 [Internet] W3C organisation. Available from <http://www.w3c.org/TR/2004/REC-xml-20040204/> [Accessed December 10th, 2004]

Clark (1999) XSL Transformations (XSLT) Version 1.0 W3C Recommendation 16 November 1999 [Internet] W3C organisation. Available from <http://www.w3c.org/TR/xslt> [Accessed December 10th, 2004]

Clark, DeRose (1999) XML Path Language (XPath) Version 1.0 W3C Recommendation 16 November 1999 [Internet] W3C organisation. Available from < http://www.w3c.org/TR/xpath> [Accessed December 10th, 2004]

Cox, Cuthbert, Lake, Martell (2002) OpenGIS Implementation Specification: OpenGIS Geography Markup Language (GML) Implemnataion Specification, version 2.1.2 [Internet] Open Geospatial Consortium. Available from: <http://www.opengeospatial.org/docs/02-069.pdf> [Accessed November 8th, 2004]

Deegree Homepage [Internet] <http://deegree.sourceforge.net> [Accessed Jannuary 23th, 2005]> [Accessed October 25th, 2005]

Evans (2003) OpenGIS Implementation Specification: Web Coverage Service (WCS), Version 1.0.0 [Internet] Open Geospatial Consortium. Available from: <http://portal.opengeospatial.org/files?artifact_id=3837> [Accessed November 8th, 2004]

GeoServer Homepage [Internet] <http://geoserver.sourceforge.net/html/index.php> [Accessed Jannuary 23th, 2005]> [Accessed October 25th, 2005]

Grässle, Baumann (2003) UML projektorientiert. 2. Auflage. Paderborn BRD, Gallilio Computing

Håkon, Bos (1999) Cascading Style Sheets, level 1: W3C Recommendation 17 Dec 1996, revised 11 Jan 1999 [Internet] W3C organisation. Available from http://www.w3c.org/TR/CSS1> [Accessed December 10th, 2004]

Java Homepage [Internet] Sun Microsystems <http://java.sun.com> [Accessed Jannuary 23th, 2005]> [Accessed October 25th, 2005]

Java Tutorial [Internet] Sun Microsystems <http://java.sun.com/docs/books/tutorial/> [Accessed Jannuary 23th, 2005]> [Accessed October 25th, 2005]

Percivall (2002) The OpenGIS Abstract Specification Topic 12: OpenGIS Service Architecture [Internet] Open Geospatial Consortium. Available from: <http://www.opengis.org/docs/02-112.pdf> [Accesed December 23th 2004]

Lake, Burggraf, Trninic, Rae (2004) GML Geographic Mark-Up Language, foundation for the Geo-Web. Chichester - West Sussex England, John Wiley & Sons, Ltd

Raggett, Le Hors, Jacobs (1999) HTML 4.01 Specification W3C Recommendation 24 December 1999 [Internet] W3C organisation. Available from <http://www.w3.org/TR/html4/> [Accessed December 10th, 2004]

Skulschus, Wiederstein (2004): XML Schema. Bonn BRD, Gallilieo Press GmbH

Velocity Homepage [Internet] The Apache Software Foundation <http://jakarta.apache.org/velocity/index.html> [Accessed Jannuary 23th, 2005]> [Accessed October 25th, 2005]

Vretanos (2002) OpenGIS Implementation Specification: Web Feature Implementation Specification, Version 1.0.0 [Internet] Open Geospatial Consortium. Available from: <http://portal.opengeospatial.org/files?artifact_id=7176> [Accessed November 8th, 2004]

Appendix

Abbreviations

CSS	Cascading Stylesheet
DPI	Dots per inch
GIS	Geographic Information System
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
HTTP GET	HTTP request where attributes are provided as key value pairs
HTTP POST	HTTP request where request is provided inline in request body
IP	Information Product
J2EE	Java 2 Enterprise Edition version
JAVA	Programming language
JSP	Java Server Page
OGC	Open Geospatial Consortium
OWS	OGC Webservice
OWS	Open Web Service
SLD	Styled Layer Descriptor
SOA	Service Orientated Architecture
SRS	Spatial Refernece System
UML	Unified modelling language
URL	Uniform Resource Locator
WCS	Web Coverage Service
WFS	Web Feature Service
WMC	Web Map Context Document
WMC	Web Map Context document
WMS	Web Mapping Service
XML	Extensible Markup Language
XPath	Query language for XML documents
XSL	Extensible Stylesheet Language

Glossary of GIS functions

The glossary of GIS functions was derived from the Lexicon of GIS Functions (Tomlinson, 2003).

Basic system capabilities

Data input

Keyboard input	Process of putting manually alphanumeric data in a machine readable form.
Edit and display (on input)	Editing and displaying geographic data are necessary to visualize data, correct errors, and add data to already digitised data.
Add attributes	Adds descriptive alphanumeric data to a digital map or an existing attribute table.

Storing, maintaining, and outputting data

Edit and display (on output)	Is needed to create effective map products. This requires a variety of editing, layout, symbolisation and plotting functions.
Symbolize	Is the process of selecting and using a variety of symbols to represent features as printed output as well as on the display.
Plot	Allows creating a hardcopy output on a printer or a plotter.
Browse	Allows identifying and defining an area of interest which is further on used by other functions.
Suppress	Removes features from the working environment. They are omitted in the subsequent manipulation and analysis processes.
Create list (report)	Create information products in the form of tables, list and or reports.

Data manipulation and analysis functions

Query

Spatial query	Selects a subset of a dataset based on spatial characteristics.
Attribute query	Selects a subset of a dataset based on their attributes.

Generating features, views, and graphs

Generate buffer	Generates a zone around a point, line or area features.
Generate graph	Is the ability to generate a graph from attribute data. Usually they display two attributes as x and y axis.

Manipulating features

Classify attributes	Groups features with similar values or attributes into classes.
Clip	Extract features from defined areas (commonly referred as cookie-cutting).
Scale change	Changes the size in which the data is displayed. Can only be performed in the computer.
Projection change	Can be necessary by integrating datasets from another source.

Address locations

Address match	Is the ability to match addresses that identify the same place recorded in different ways.
Address geocode (search)	Allows to ad point locations derived from address information to a map. This function is extended by the author. He understands also the search and location of features as geocoding.

Measurement

Measure length	Allows to measure the length of a line
Measure perimeter	Allows to measure the perimeter of an area
Measure area	Ability to measure the area of a polygon.

Calculation

Arithmetic calculation	Perform operations like addition, subtraction, multiplication and division.
Algebraic calculation	Ability to perform operations on logical expressions.
Statistical calculation	Perform statistical analysis and tests.

Spatial Analysis

Graphic overplot	Allows superimposing one map on another, to see the intersections of the datasets.
Topological overlay	Produces new data as result of the combination of the two input data layers.
Adjacency analysis	Identifies areas that are next (adjacent) to each other.
Nearest neighbour search	Identifies individual or sets of features that are nearest to other features specified by location or attributes.
Correlation analysis	Compare two maps of the same area, which represent condition of different time periods.

Interfaces

- interface java.lang.Cloneable
  - interface net.opengis.context.**Abstract** (also extends java.io.Serializable)
  - interface net.opengis.sld.**Abstract** (also extends java.io.Serializable)
  - interface net.opengis.gml.**AbstractGeometryCollectionBaseType** (also extends java.io.Serializable)
  - interface net.opengis.gml.**AbstractGeometryType** (also extends java.io.Serializable)
    - interface net.opengis.gml.**_Geometry**
      - interface net.opengis.gml.**_GeometryCollection**
      - interface net.opengis.gml.**LinearRing**
      - interface net.opengis.gml.**LineString**
      - interface net.opengis.gml.**MultiGeometry**
      - interface net.opengis.gml.**MultiLineString**
      - interface net.opengis.gml.**MultiPoint**
      - interface net.opengis.gml.**MultiPolygon**
      - interface net.opengis.gml.**Point**
      - interface net.opengis.gml.**Polygon**
  - interface net.opengis.context.**Address** (also extends java.io.Serializable)
  - interface net.opengis.context.**AddressType** (also extends java.io.Serializable)
    - interface net.opengis.context.**ContactAddress**
  - interface net.opengis.sld.**AnchorPoint** (also extends java.io.Serializable)
  - interface net.opengis.sld.**AVERAGE** (also extends java.io.Serializable)
  - interface net.opengis.ogc.**BBOXType** (also extends java.io.Serializable)
  - interface net.opengis.ogc.**BinaryComparisonOpType** (also extends java.io.Serializable)
  - interface net.opengis.ogc.**BinaryLogicOpType** (also extends java.io.Serializable)
  - interface net.opengis.ogc.**BinaryOperatorType** (also extends java.io.Serializable)
  - interface net.opengis.ogc.**BinarySpatialOpType** (also extends java.io.Serializable)
  - interface net.opengis.context.**BoundingBoxType** (also extends java.io.Serializable)
    - interface net.opengis.context.**BoundingBox**
  - interface net.opengis.gml.**BoxType** (also extends java.io.Serializable)
    - interface net.opengis.gml.**Box**
  - interface net.opengis.sld.**BrightnessOnly** (also extends java.io.Serializable)
  - interface net.opengis.sld.**ChannelSelection** (also extends java.io.Serializable)
  - interface net.opengis.context.**City** (also extends java.io.Serializable)
  - interface net.opengis.sld.**ColorMap** (also extends java.io.Serializable)
  - interface net.opengis.sld.**ColorMapEntry** (also extends java.io.Serializable)
  - interface net.opengis.ogc.**ComparisonOpsType** (also extends java.io.Serializable)
    - interface net.opengis.ogc.**ComparisonOps**
      - interface net.opengis.ogc.**PropertyIsBetween**
      - interface net.opengis.ogc.**PropertyIsEqualTo**
      - interface net.opengis.ogc.**PropertyIsGreaterThan**
      - interface net.opengis.ogc.**PropertyIsGreaterThanOrEqualTo**
      - interface net.opengis.ogc.**PropertyIsLessThan**
      - interface net.opengis.ogc.**PropertyIsLessThanOrEqualTo**
      - interface net.opengis.ogc.**PropertyIsLike**
      - interface net.opengis.ogc.**PropertyIsNotEqualTo**
      - interface net.opengis.ogc.**PropertyIsNull**
  - interface net.opengis.context.**ContactElectronicMailAddress** (also extends java.io.Serializable)
  - interface net.opengis.context.**ContactFacsimileTelephone** (also extends java.io.Serializable)
  - interface net.opengis.context.**ContactInformationType** (also extends java.io.Serializable)
    - interface net.opengis.context.**ContactInformation**
  - interface net.opengis.context.**ContactOrganization** (also extends java.io.Serializable)
  - interface net.opengis.context.**ContactPerson** (also extends java.io.Serializable)
  - interface net.opengis.context.**ContactPersonPrimaryType** (also extends

java.io.Serializable)
    - interface net.opengis.context.**ContactPersonPrimary**
- interface net.opengis.context.**ContactPosition** (also extends java.io.Serializable)
- interface net.opengis.context.**ContactVoiceTelephone** (also extends java.io.Serializable)
- interface net.opengis.context.**ContextURLType** (also extends java.io.Serializable)
    - interface net.opengis.context.**ContextURL**
- interface net.opengis.sld.**ContrastEnhancement** (also extends java.io.Serializable)
- interface net.opengis.gml.**CoordinatesType** (also extends java.io.Serializable)
    - interface net.opengis.gml.**Coordinates**
- interface net.opengis.gml.**CoordType** (also extends java.io.Serializable)
    - interface net.opengis.gml.**Coord**
- interface net.opengis.context.**Country** (also extends java.io.Serializable)
- interface net.opengis.sld.**CssParameter** (also extends java.io.Serializable)
- interface net.opengis.sld.**Displacement** (also extends java.io.Serializable)
- interface net.opengis.ogc.**DistanceBufferType** (also extends java.io.Serializable)
- interface net.opengis.ogc.**DistanceType** (also extends java.io.Serializable)
    - interface net.opengis.ogc.**Distance**
- interface net.opengis.sld.**EARLIEST_ON_TOP** (also extends java.io.Serializable)
- interface net.opengis.sld.**ElseFilter** (also extends java.io.Serializable)
- interface net.opengis.ogc.**ExpressionType** (also extends java.io.Serializable)
    - interface net.opengis.ogc.**Expression**
        - interface net.opengis.ogc.**Add**
        - interface net.opengis.ogc.**Div**
        - interface net.opengis.ogc.**Function**
        - interface net.opengis.ogc.**Literal**
        - interface net.opengis.ogc.**Mul**
        - interface net.opengis.ogc.**PropertyName**
        - interface net.opengis.ogc.**Sub**
- interface net.opengis.context.**ExtensionType** (also extends java.io.Serializable)
    - interface net.opengis.context.**Extension**
- interface net.opengis.sld.**Extent** (also extends java.io.Serializable)
- interface net.opengis.sld.**ExternalGraphic** (also extends java.io.Serializable)
- interface net.opengis.ogc.**FeatureIdType** (also extends java.io.Serializable)
    - interface net.opengis.ogc.**FeatureId**
- interface net.opengis.sld.**FeatureTypeConstraint** (also extends java.io.Serializable)
- interface net.opengis.sld.**FeatureTypeName** (also extends java.io.Serializable)
- interface net.opengis.sld.**FeatureTypeStyle** (also extends java.io.Serializable)
- interface net.opengis.sld.**Fill** (also extends java.io.Serializable)
- interface net.opengis.ogc.**FilterType** (also extends java.io.Serializable)
    - interface net.opengis.ogc.**Filter**
- interface net.opengis.sld.**Font** (also extends java.io.Serializable)
- interface net.opengis.sld.**Format** (also extends java.io.Serializable)
- interface net.opengis.context.**FormatListType** (also extends java.io.Serializable)
    - interface net.opengis.context.**FormatList**
- interface net.opengis.context.**FormatType** (also extends java.io.Serializable)
    - interface net.opengis.context.**Format**
- interface net.opengis.ogc.**FunctionType** (also extends java.io.Serializable)
- interface net.opengis.sld.**GammaValue** (also extends java.io.Serializable)
- interface net.opengis.context.**GeneralType** (also extends java.io.Serializable)
    - interface net.opengis.context.**General**
- interface net.opengis.sld.**Geometry** (also extends java.io.Serializable)
- interface net.opengis.gml.**GeometryAssociationType** (also extends java.io.Serializable)
    - interface net.opengis.gml.**GeometryMember**
        - interface net.opengis.gml.**LineStringMember**
        - interface net.opengis.gml.**PointMember**
        - interface net.opengis.gml.**PolygonMember**
- interface net.opengis.gml.**GeometryCollectionType** (also extends java.io.Serializable)
- interface net.opengis.sld.**Graphic** (also extends java.io.Serializable)

- interface net.opengis.sld.**GraphicFill** (also extends java.io.Serializable)
- interface net.opengis.sld.**GraphicStroke** (also extends java.io.Serializable)
- interface net.opengis.sld.**Halo** (also extends java.io.Serializable)
- interface net.opengis.sld.**Histogram** (also extends java.io.Serializable)
- interface net.opengis.sld.**ImageOutline** (also extends java.io.Serializable)
- interface net.opengis.sld.**IsDefault** (also extends java.io.Serializable)
- interface net.opengis.context.**Keyword** (also extends java.io.Serializable)
- interface net.opengis.context.**KeywordListType** (also extends java.io.Serializable)
  - interface net.opengis.context.**KeywordList**
- interface net.opengis.sld.**LabelPlacement** (also extends java.io.Serializable)
- interface net.opengis.sld.**LATEST_ON_TOP** (also extends java.io.Serializable)
- interface net.opengis.sld.**LayerFeatureConstraints** (also extends java.io.Serializable)
- interface net.opengis.context.**LayerListType** (also extends java.io.Serializable)
  - interface net.opengis.context.**LayerList**
- interface net.opengis.context.**LayerType** (also extends java.io.Serializable)
  - interface net.opengis.context.**Layer**
- interface net.opengis.sld.**LegendGraphic** (also extends java.io.Serializable)
- interface net.opengis.gml.**LinearRingMemberType** (also extends java.io.Serializable)
  - interface net.opengis.gml.**InnerBoundaryIs**
  - interface net.opengis.gml.**OuterBoundaryIs**
- interface net.opengis.gml.**LinearRingType** (also extends java.io.Serializable)
- interface net.opengis.sld.**LinePlacement** (also extends java.io.Serializable)
- interface net.opengis.gml.**LineStringMemberType** (also extends java.io.Serializable)
- interface net.opengis.gml.**LineStringType** (also extends java.io.Serializable)
- interface net.opengis.ogc.**LiteralType** (also extends java.io.Serializable)
- interface net.opengis.ogc.**LogicOpsType** (also extends java.io.Serializable)
  - interface net.opengis.ogc.**LogicOps**
    - interface net.opengis.ogc.**And**
    - interface net.opengis.ogc.**Not**
    - interface net.opengis.ogc.**Or**
- interface net.opengis.ogc.**LowerBoundaryType** (also extends java.io.Serializable)
  - interface net.opengis.ogc.**LowerBoundary**
- interface net.opengis.sld.**Mark** (also extends java.io.Serializable)
- interface net.opengis.sld.**MaxScaleDenominator** (also extends java.io.Serializable)
- interface net.opengis.sld.**MinScaleDenominator** (also extends java.io.Serializable)
- interface net.opengis.gml.**MultiLineStringType** (also extends java.io.Serializable)
- interface net.opengis.gml.**MultiPointType** (also extends java.io.Serializable)
- interface net.opengis.gml.**MultiPolygonType** (also extends java.io.Serializable)
- interface net.opengis.context.**Name** (also extends java.io.Serializable)
- interface net.opengis.sld.**Name** (also extends java.io.Serializable)
- interface net.opengis.sld.**NamedLayer** (also extends java.io.Serializable)
- interface net.opengis.sld.**NamedStyle** (also extends java.io.Serializable)
- interface net.opengis.sld.**Normalize** (also extends java.io.Serializable)
- interface net.opengis.sld.**OnlineResource** (also extends java.io.Serializable)
- interface net.opengis.context.**OnlineResourceType** (also extends java.io.Serializable)
  - interface net.opengis.context.**OnlineResource**
- interface net.opengis.sld.**OverlapBehavior** (also extends java.io.Serializable)
- interface net.opengis.sld.**ParameterValueType** (also extends java.io.Serializable)
  - interface net.opengis.sld.**AnchorPointX**
  - interface net.opengis.sld.**AnchorPointY**
  - interface net.opengis.sld.**DisplacementX**
  - interface net.opengis.sld.**DisplacementY**
  - interface net.opengis.sld.**Label**
  - interface net.opengis.sld.**Opacity**
  - interface net.opengis.sld.**PerpendicularOffset**
  - interface net.opengis.sld.**Radius**
  - interface net.opengis.sld.**Rotation**
  - interface net.opengis.sld.**Size**
- interface net.opengis.gml.**PointMemberType** (also extends java.io.Serializable)
- interface net.opengis.sld.**PointPlacement** (also extends java.io.Serializable)

- interface net.opengis.gml.**PointType** (also extends java.io.Serializable)
- interface net.opengis.gml.**PolygonMemberType** (also extends java.io.Serializable)
- interface net.opengis.gml.**PolygonType** (also extends java.io.Serializable)
- interface net.opengis.context.**PostCode** (also extends java.io.Serializable)
- interface net.opengis.ogc.**PropertyIsBetweenType** (also extends java.io.Serializable)
- interface net.opengis.ogc.**PropertyIsLikeType** (also extends java.io.Serializable)
- interface net.opengis.ogc.**PropertyIsNullType** (also extends java.io.Serializable)
- interface net.opengis.ogc.**PropertyNameType** (also extends java.io.Serializable)
- interface net.opengis.sld.**RANDOM** (also extends java.io.Serializable)
- interface net.opengis.sld.**ReliefFactor** (also extends java.io.Serializable)
- interface net.opengis.sld.**RemoteOWS** (also extends java.io.Serializable)
- interface net.opengis.sld.**Rule** (also extends java.io.Serializable)
- interface net.opengis.sld.**SelectedChannelType** (also extends java.io.Serializable)
  - interface net.opengis.sld.**BlueChannel**
  - interface net.opengis.sld.**GrayChannel**
  - interface net.opengis.sld.**GreenChannel**
  - interface net.opengis.sld.**RedChannel**
- interface net.opengis.sld.**SemanticTypeIdentifier** (also extends java.io.Serializable)
- interface net.opengis.context.**ServerType** (also extends java.io.Serializable)
  - interface net.opengis.context.**Server**
- interface net.opengis.sld.**Service** (also extends java.io.Serializable)
- interface net.opengis.sld.**ShadedRelief** (also extends java.io.Serializable)
- interface net.opengis.context.**SLDType** (also extends java.io.Serializable)
  - interface net.opengis.context.**SLD**
- interface net.opengis.sld.**SourceChannelName** (also extends java.io.Serializable)
- interface net.opengis.ogc.**SpatialOpsType** (also extends java.io.Serializable)
  - interface net.opengis.ogc.**SpatialOps**
    - interface net.opengis.ogc.**BBOX**
    - interface net.opengis.ogc.**Beyond**
    - interface net.opengis.ogc.**Contains**
    - interface net.opengis.ogc.**Crosses**
    - interface net.opengis.ogc.**Disjoint**
    - interface net.opengis.ogc.**DWithin**
    - interface net.opengis.ogc.**Equals**
    - interface net.opengis.ogc.**Intersects**
    - interface net.opengis.ogc.**Overlaps**
    - interface net.opengis.ogc.**Touches**
    - interface net.opengis.ogc.**Within**
- interface net.opengis.context.**SRS** (also extends java.io.Serializable)
- interface net.opengis.context.**StateOrProvince** (also extends java.io.Serializable)
- interface net.opengis.sld.**Stroke** (also extends java.io.Serializable)
- interface net.opengis.sld.**StyledLayerDescriptor** (also extends java.io.Serializable)
- interface net.opengis.context.**StyleListType** (also extends java.io.Serializable)
  - interface net.opengis.context.**StyleList**
- interface net.opengis.context.**StyleType** (also extends java.io.Serializable)
  - interface net.opengis.context.**Style**
- interface net.opengis.sld.**SymbolizerType** (also extends java.io.Serializable)
  - interface net.opengis.sld.**Symbolizer**
    - interface net.opengis.sld.**LineSymbolizer**
    - interface net.opengis.sld.**PointSymbolizer**
    - interface net.opengis.sld.**PolygonSymbolizer**
    - interface net.opengis.sld.**RasterSymbolizer**
    - interface net.opengis.sld.**TextSymbolizer**
- interface net.opengis.context.**Title** (also extends java.io.Serializable)
- interface net.opengis.sld.**Title** (also extends java.io.Serializable)
- interface net.opengis.ogc.**UnaryLogicOpType** (also extends java.io.Serializable)
- interface net.opengis.ogc.**UpperBoundaryType** (also extends java.io.Serializable)
  - interface net.opengis.ogc.**UpperBoundary**
- interface net.opengis.context.**URLType** (also extends java.io.Serializable)
  - interface net.opengis.context.**DataURL**

- interface net.opengis.context.**DescriptionURL**
- interface net.opengis.context.**LegendURL**
- interface net.opengis.context.**LogoURL**
- interface net.opengis.context.**MetadataURL**
- interface net.opengis.sld.**UserLayer** (also extends java.io.Serializable)
- interface net.opengis.sld.**UserStyle** (also extends java.io.Serializable)
- interface net.opengis.sld.**Value** (also extends java.io.Serializable)
- interface net.opengis.context.**ViewContextCollectionType** (also extends java.io.Serializable)
  - interface net.opengis.context.**ViewContextCollection**
- interface net.opengis.context.**ViewContextReferenceType** (also extends java.io.Serializable)
  - interface net.opengis.context.**ViewContextReference**
- interface net.opengis.context.**ViewContextType** (also extends java.io.Serializable)
  - interface net.opengis.context.**ViewContext**
- interface net.opengis.sld.**WellKnownName** (also extends java.io.Serializable)
- interface net.opengis.context.**WindowType** (also extends java.io.Serializable)
  - interface net.opengis.context.**Window**
- interface net.opengis.gml.**X** (also extends java.io.Serializable)
- interface net.opengis.gml.**Y** (also extends java.io.Serializable)
- interface net.opengis.gml.**Z** (also extends java.io.Serializable)

- interface java.io.Serializable
  - interface net.opengis.context.**Abstract** (also extends java.lang.Cloneable)
  - interface net.opengis.sld.**Abstract** (also extends java.lang.Cloneable)
  - interface net.opengis.gml.**AbstractGeometryCollectionBaseType** (also extends java.lang.Cloneable)
  - interface net.opengis.gml.**AbstractGeometryType** (also extends java.lang.Cloneable)
    - interface net.opengis.gml.**_Geometry**
      - interface net.opengis.gml.**_GeometryCollection**
      - interface net.opengis.gml.**LinearRing**
      - interface net.opengis.gml.**LineString**
      - interface net.opengis.gml.**MultiGeometry**
      - interface net.opengis.gml.**MultiLineString**
      - interface net.opengis.gml.**MultiPoint**
      - interface net.opengis.gml.**MultiPolygon**
      - interface net.opengis.gml.**Point**
      - interface net.opengis.gml.**Polygon**
  - interface net.opengis.context.**Address** (also extends java.lang.Cloneable)
  - interface net.opengis.context.**AddressType** (also extends java.lang.Cloneable)
    - interface net.opengis.context.**ContactAddress**
  - interface net.opengis.sld.**AnchorPoint** (also extends java.lang.Cloneable)
  - interface net.opengis.sld.**AVERAGE** (also extends java.lang.Cloneable)
  - interface net.opengis.ogc.**BBOXType** (also extends java.lang.Cloneable)
  - interface net.opengis.ogc.**BinaryComparisonOpType** (also extends java.lang.Cloneable)
  - interface net.opengis.ogc.**BinaryLogicOpType** (also extends java.lang.Cloneable)
  - interface net.opengis.ogc.**BinaryOperatorType** (also extends java.lang.Cloneable)
  - interface net.opengis.ogc.**BinarySpatialOpType** (also extends java.lang.Cloneable)
  - interface net.opengis.context.**BoundingBoxType** (also extends java.lang.Cloneable)
    - interface net.opengis.context.**BoundingBox**
  - interface net.opengis.gml.**BoxType** (also extends java.lang.Cloneable)
    - interface net.opengis.gml.**Box**
  - interface net.opengis.sld.**BrightnessOnly** (also extends java.lang.Cloneable)
  - interface net.opengis.sld.**ChannelSelection** (also extends java.lang.Cloneable)
  - interface net.opengis.context.**City** (also extends java.lang.Cloneable)
  - interface net.opengis.sld.**ColorMap** (also extends java.lang.Cloneable)
  - interface net.opengis.sld.**ColorMapEntry** (also extends java.lang.Cloneable)
  - interface net.opengis.ogc.**ComparisonOpsType** (also extends java.lang.Cloneable)
    - interface net.opengis.ogc.**ComparisonOps**
      - interface net.opengis.ogc.**PropertyIsBetween**

- interface net.opengis.ogc.**PropertyIsEqualTo**
- interface net.opengis.ogc.**PropertyIsGreaterThan**
- interface net.opengis.ogc.**PropertyIsGreaterThanOrEqualTo**
- interface net.opengis.ogc.**PropertyIsLessThan**
- interface net.opengis.ogc.**PropertyIsLessThanOrEqualTo**
- interface net.opengis.ogc.**PropertyIsLike**
- interface net.opengis.ogc.**PropertyIsNotEqualTo**
- interface net.opengis.ogc.**PropertyIsNull**

- interface net.opengis.context.**ContactElectronicMailAddress** (also extends java.lang.Cloneable)
- interface net.opengis.context.**ContactFacsimileTelephone** (also extends java.lang.Cloneable)
- interface net.opengis.context.**ContactInformationType** (also extends java.lang.Cloneable)
  - interface net.opengis.context.**ContactInformation**
- interface net.opengis.context.**ContactOrganization** (also extends java.lang.Cloneable)
- interface net.opengis.context.**ContactPerson** (also extends java.lang.Cloneable)
- interface net.opengis.context.**ContactPersonPrimaryType** (also extends java.lang.Cloneable)
  - interface net.opengis.context.**ContactPersonPrimary**
- interface net.opengis.context.**ContactPosition** (also extends java.lang.Cloneable)
- interface net.opengis.context.**ContactVoiceTelephone** (also extends java.lang.Cloneable)
- interface net.opengis.context.**ContextURLType** (also extends java.lang.Cloneable)
  - interface net.opengis.context.**ContextURL**
- interface net.opengis.sld.**ContrastEnhancement** (also extends java.lang.Cloneable)
- interface net.opengis.gml.**CoordinatesType** (also extends java.lang.Cloneable)
  - interface net.opengis.gml.**Coordinates**
- interface net.opengis.gml.**CoordType** (also extends java.lang.Cloneable)
  - interface net.opengis.gml.**Coord**
- interface net.opengis.context.**Country** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**CssParameter** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**Displacement** (also extends java.lang.Cloneable)
- interface net.opengis.ogc.**DistanceBufferType** (also extends java.lang.Cloneable)
- interface net.opengis.ogc.**DistanceType** (also extends java.lang.Cloneable)
  - interface net.opengis.ogc.**Distance**
- interface net.opengis.sld.**EARLIEST_ON_TOP** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**ElseFilter** (also extends java.lang.Cloneable)
- interface net.opengis.ogc.**ExpressionType** (also extends java.lang.Cloneable)
  - interface net.opengis.ogc.**Expression**
    - interface net.opengis.ogc.**Add**
    - interface net.opengis.ogc.**Div**
    - interface net.opengis.ogc.**Function**
    - interface net.opengis.ogc.**Literal**
    - interface net.opengis.ogc.**Mul**
    - interface net.opengis.ogc.**PropertyName**
    - interface net.opengis.ogc.**Sub**
- interface net.opengis.context.**ExtensionType** (also extends java.lang.Cloneable)
  - interface net.opengis.context.**Extension**
- interface net.opengis.sld.**Extent** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**ExternalGraphic** (also extends java.lang.Cloneable)
- interface net.opengis.ogc.**FeatureIdType** (also extends java.lang.Cloneable)
  - interface net.opengis.ogc.**FeatureId**
- interface net.opengis.sld.**FeatureTypeConstraint** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**FeatureTypeName** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**FeatureTypeStyle** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**Fill** (also extends java.lang.Cloneable)
- interface net.opengis.ogc.**FilterType** (also extends java.lang.Cloneable)
  - interface net.opengis.ogc.**Filter**
- interface net.opengis.sld.**Font** (also extends java.lang.Cloneable)

- interface net.opengis.sld.**Format** (also extends java.lang.Cloneable)
- interface net.opengis.context.**FormatListType** (also extends java.lang.Cloneable)
  - interface net.opengis.context.**FormatList**
- interface net.opengis.context.**FormatType** (also extends java.lang.Cloneable)
  - interface net.opengis.context.**Format**
- interface net.opengis.ogc.**FunctionType** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**GammaValue** (also extends java.lang.Cloneable)
- interface net.opengis.context.**GeneralType** (also extends java.lang.Cloneable)
  - interface net.opengis.context.**General**
- interface net.opengis.sld.**Geometry** (also extends java.lang.Cloneable)
- interface net.opengis.gml.**GeometryAssociationType** (also extends java.lang.Cloneable)
  - interface net.opengis.gml.**GeometryMember**
    - interface net.opengis.gml.**LineStringMember**
    - interface net.opengis.gml.**PointMember**
    - interface net.opengis.gml.**PolygonMember**
- interface net.opengis.gml.**GeometryCollectionType** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**Graphic** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**GraphicFill** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**GraphicStroke** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**Halo** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**Histogram** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**ImageOutline** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**IsDefault** (also extends java.lang.Cloneable)
- interface net.opengis.context.**Keyword** (also extends java.lang.Cloneable)
- interface net.opengis.context.**KeywordListType** (also extends java.lang.Cloneable)
  - interface net.opengis.context.**KeywordList**
- interface net.opengis.sld.**LabelPlacement** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**LATEST_ON_TOP** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**LayerFeatureConstraints** (also extends java.lang.Cloneable)
- interface net.opengis.context.**LayerListType** (also extends java.lang.Cloneable)
  - interface net.opengis.context.**LayerList**
- interface net.opengis.context.**LayerType** (also extends java.lang.Cloneable)
  - interface net.opengis.context.**Layer**
- interface net.opengis.sld.**LegendGraphic** (also extends java.lang.Cloneable)
- interface net.opengis.gml.**LinearRingMemberType** (also extends java.lang.Cloneable)
  - interface net.opengis.gml.**InnerBoundaryIs**
  - interface net.opengis.gml.**OuterBoundaryIs**
- interface net.opengis.gml.**LinearRingType** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**LinePlacement** (also extends java.lang.Cloneable)
- interface net.opengis.gml.**LineStringMemberType** (also extends java.lang.Cloneable)
- interface net.opengis.gml.**LineStringType** (also extends java.lang.Cloneable)
- interface net.opengis.ogc.**LiteralType** (also extends java.lang.Cloneable)
- interface net.opengis.ogc.**LogicOpsType** (also extends java.lang.Cloneable)
  - interface net.opengis.ogc.**LogicOps**
    - interface net.opengis.ogc.**And**
    - interface net.opengis.ogc.**Not**
    - interface net.opengis.ogc.**Or**
- interface net.opengis.ogc.**LowerBoundaryType** (also extends java.lang.Cloneable)
  - interface net.opengis.ogc.**LowerBoundary**
- interface net.opengis.sld.**Mark** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**MaxScaleDenominator** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**MinScaleDenominator** (also extends java.lang.Cloneable)
- interface net.opengis.gml.**MultiLineStringType** (also extends java.lang.Cloneable)
- interface net.opengis.gml.**MultiPointType** (also extends java.lang.Cloneable)
- interface net.opengis.gml.**MultiPolygonType** (also extends java.lang.Cloneable)
- interface net.opengis.context.**Name** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**Name** (also extends java.lang.Cloneable)

- interface net.opengis.sld.**NamedLayer** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**NamedStyle** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**Normalize** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**OnlineResource** (also extends java.lang.Cloneable)
- interface net.opengis.context.**OnlineResourceType** (also extends java.lang.Cloneable)
  - interface net.opengis.context.**OnlineResource**
- interface net.opengis.sld.**OverlapBehavior** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**ParameterValueType** (also extends java.lang.Cloneable)
  - interface net.opengis.sld.**AnchorPointX**
  - interface net.opengis.sld.**AnchorPointY**
  - interface net.opengis.sld.**DisplacementX**
  - interface net.opengis.sld.**DisplacementY**
  - interface net.opengis.sld.**Label**
  - interface net.opengis.sld.**Opacity**
  - interface net.opengis.sld.**PerpendicularOffset**
  - interface net.opengis.sld.**Radius**
  - interface net.opengis.sld.**Rotation**
  - interface net.opengis.sld.**Size**
- interface net.opengis.gml.**PointMemberType** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**PointPlacement** (also extends java.lang.Cloneable)
- interface net.opengis.gml.**PointType** (also extends java.lang.Cloneable)
- interface net.opengis.gml.**PolygonMemberType** (also extends java.lang.Cloneable)
- interface net.opengis.gml.**PolygonType** (also extends java.lang.Cloneable)
- interface net.opengis.context.**PostCode** (also extends java.lang.Cloneable)
- interface net.opengis.ogc.**PropertyIsBetweenType** (also extends java.lang.Cloneable)
- interface net.opengis.ogc.**PropertyIsLikeType** (also extends java.lang.Cloneable)
- interface net.opengis.ogc.**PropertyIsNullType** (also extends java.lang.Cloneable)
- interface net.opengis.ogc.**PropertyNameType** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**RANDOM** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**ReliefFactor** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**RemoteOWS** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**Rule** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**SelectedChannelType** (also extends java.lang.Cloneable)
  - interface net.opengis.sld.**BlueChannel**
  - interface net.opengis.sld.**GrayChannel**
  - interface net.opengis.sld.**GreenChannel**
  - interface net.opengis.sld.**RedChannel**
- interface net.opengis.sld.**SemanticTypeIdentifier** (also extends java.lang.Cloneable)
- interface net.opengis.context.**ServerType** (also extends java.lang.Cloneable)
  - interface net.opengis.context.**Server**
- interface net.opengis.sld.**Service** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**ShadedRelief** (also extends java.lang.Cloneable)
- interface net.opengis.context.**SLDType** (also extends java.lang.Cloneable)
  - interface net.opengis.context.**SLD**
- interface net.opengis.sld.**SourceChannelName** (also extends java.lang.Cloneable)
- interface net.opengis.ogc.**SpatialOpsType** (also extends java.lang.Cloneable)
  - interface net.opengis.ogc.**SpatialOps**
    - interface net.opengis.ogc.**BBOX**
    - interface net.opengis.ogc.**Beyond**
    - interface net.opengis.ogc.**Contains**
    - interface net.opengis.ogc.**Crosses**
    - interface net.opengis.ogc.**Disjoint**
    - interface net.opengis.ogc.**DWithin**
    - interface net.opengis.ogc.**Equals**
    - interface net.opengis.ogc.**Intersects**
    - interface net.opengis.ogc.**Overlaps**
    - interface net.opengis.ogc.**Touches**
    - interface net.opengis.ogc.**Within**
- interface net.opengis.context.**SRS** (also extends java.lang.Cloneable)
- interface net.opengis.context.**StateOrProvince** (also extends java.lang.Cloneable)

- interface net.opengis.sld.**Stroke** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**StyledLayerDescriptor** (also extends java.lang.Cloneable)
- interface net.opengis.context.**StyleListType** (also extends java.lang.Cloneable)
  - interface net.opengis.context.**StyleList**
- interface net.opengis.context.**StyleType** (also extends java.lang.Cloneable)
  - interface net.opengis.context.**Style**
- interface net.opengis.sld.**SymbolizerType** (also extends java.lang.Cloneable)
  - interface net.opengis.sld.**Symbolizer**
    - interface net.opengis.sld.**LineSymbolizer**
    - interface net.opengis.sld.**PointSymbolizer**
    - interface net.opengis.sld.**PolygonSymbolizer**
    - interface net.opengis.sld.**RasterSymbolizer**
    - interface net.opengis.sld.**TextSymbolizer**
- interface net.opengis.context.**Title** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**Title** (also extends java.lang.Cloneable)
- interface net.opengis.ogc.**UnaryLogicOpType** (also extends java.lang.Cloneable)
- interface net.opengis.ogc.**UpperBoundaryType** (also extends java.lang.Cloneable)
  - interface net.opengis.ogc.**UpperBoundary**
- interface net.opengis.context.**URLType** (also extends java.lang.Cloneable)
  - interface net.opengis.context.**DataURL**
  - interface net.opengis.context.**DescriptionURL**
  - interface net.opengis.context.**LegendURL**
  - interface net.opengis.context.**LogoURL**
  - interface net.opengis.context.**MetadataURL**
- interface net.opengis.sld.**UserLayer** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**UserStyle** (also extends java.lang.Cloneable)
- interface net.opengis.sld.**Value** (also extends java.lang.Cloneable)
- interface net.opengis.context.**ViewContextCollectionType** (also extends java.lang.Cloneable)
  - interface net.opengis.context.**ViewContextCollection**
- interface net.opengis.context.**ViewContextReferenceType** (also extends java.lang.Cloneable)
  - interface net.opengis.context.**ViewContextReference**
- interface net.opengis.context.**ViewContextType** (also extends java.lang.Cloneable)
  - interface net.opengis.context.**ViewContext**
- interface net.opengis.sld.**WellKnownName** (also extends java.lang.Cloneable)
- interface net.opengis.context.**WindowType** (also extends java.lang.Cloneable)
  - interface net.opengis.context.**Window**
- interface net.opengis.gml.**X** (also extends java.lang.Cloneable)
- interface net.opengis.gml.**Y** (also extends java.lang.Cloneable)
- interface net.opengis.gml.**Z** (also extends java.lang.Cloneable)

Figure 164 Created interfaces (API)

Tag files

Basic System capabilities

Data input

ViewContext

The ViewContext tag loads the content of a “ViewContext” document either from the file system or an URL. Behind the scenes the “ViewContext” document is parsed into a form that allows further processing. This first example reads from an URL, the second one from the file system.

```
<context:ViewContext url="The URL">
        The ViewContext  can be accessed with the ${viewContext}variable
</context:ViewContext>
```

Figure 165 Example: read from URL

```
<context:ViewContext file="The file">
        The ViewContext  can be accessed with the ${viewContext}variable
</context:ViewContext>
```

Figure 166 Example: read from file

```
<%@tag description = "put the tag description here" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%-- Attributes --%>
<%@attribute name="file" type="java.lang.String" required="false" description="Reads from a given file"%>
<%@attribute name="url" type="java.lang.String" required="false" description="Reads from a given URL"%>
<%-- Variable that stores the ViewContext --%>
<%@variable name-given="viewContext" variable-class="net.opengis.context.ViewContext" scope="NESTED"%>
<c:choose>
   <c:when test="${file!=null}">
      <c:set var="viewContext" value="<%=com.hadrbolec.ogc.context.io.ViewContextIO.read(new java.io.FileReader(new java.io.File(file)))%>"/>
   </c:when>
   <c:when test="${url!=null}">
      <c:set var="viewContext" value="<%=com.hadrbolec.ogc.context.io.ViewContextIO.read(new java.io.InputStreamReader((new java.net.URL(url)).openStream()))%>"/>
   </c:when>
</c:choose>
<jsp:doBody/>
```

Figure 167 Tag file: ViewContext.tag

Storing, maintaining, and outputting data

ViewContextWriter

The mapping mechanism for the WMC schema generates automatically a “Reader” and “Writer” class for each namespace. The author implemented a tag that writes a given ViewContext with a provided filename “The filename” into the filesystem.

```
<context:ViewContextWriter file=”The filename” viewContext=”The ViewContext”/>
```

Figure 168 Example: write ViewContext

```
<%@tag description = "Writes a ViewContext into the filesystem" pageEncoding="UTF-8"  body-content="empty"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@attribute name="file" type="java.lang.String" required="true" description="Writes to the file"%>
<%@attribute name="viewContext" type="net.opengis.context.ViewContext" required="true" description="The view context to be written"%>
<c:if test="${file!=null && viewContext!=null}">
<%com.hadrbolec.ogc.context.io.ViewContextIO.write(viewContext, new java.io.FileWriter(file));%>
</c:if>
```

Figure 169 Tag file: ViewContextWriter.tag

Edit and display (on output)

Visibility

DisplayLayer

The “DisplayLayer” tag sets the value of the hidden attribute of the respective Layer element in the “ViewContext” document to true.

```
<context:DisplayLayer layerName="The name of the layer" viewContext=”The ViewContext”>
```

Figure 170 Example: display layer

```
<%@tag description = "Displays the layer" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@attribute name="layer" type="java.lang.String" required="true"%>
<%@attribute name="viewContext" type="net.opengis.context.ViewContext" required="true"%>
<%com.hadrbolec.ogc.context.util.VisibleUtil.display(layer,viewContext);%>
```

Figure 171 Tag file: DisplayLayer.tag

HideLayer

The “HideLayer” tag sets the value of the hidden attribute of the respective Layer element in the “ViewContext” document to true.

```
<context:HideLayer layerName="The name of the layer" viewContext="The ViewContext">
```

Figure 172 example: hide layer

```
<%@tag description = "Hides the layer" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@attribute name="layer" type="java.lang.String" required="true"%>
<%@attribute name="viewContext" type="net.opengis.context.ViewContext" required="true"%>
<%com.hadrbolec.ogc.context.util.VisibleUtil.hide(layer,viewContext);%>
```

Figure 173 Tag file: HideLayer.tag

Add / remove layer

RemoveLayer

This tag removes a layer permanently from the "ViewContext".

```
<context:RemoveLayer layerName="The layer name" viewContext="The ViewContext"/>
```

Figure 174 Example: remove layer

```
<%@tag description = "put the tag description here" pageEncoding="UTF-8" body-
content="EMPTY"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@attribute name="layerName" type="java.lang.String" required="true" description="The name of
the layer"%>
<%@attribute name="viewContext" type="net.opengis.context.ViewContext" required="true"
description="The ViewContext"%>
<%com.hadrbolec.ogc.context.util.LayerUtil.remove(layer,viewContext);%>
```

Figure 175 Tag file: RemoveLayer.tag

AddLayer

This tag adds a new "Layer" to the "ViewContext" document. The content between the "AddLayer" tag is parsed into a "Layer" element. In the case a "Layer" with the same name exists it is substituted with the new one. Otherwise the new layer is added after the last one. This means the layer is displayed on the top position.

```
<context:AddLayer viewContext="The ViewContext"> The layer XML fragment</context:AddLayer>
```

Figure 176 example: add layer

```
<%@tag description = "put the tag description here" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@attribute name="viewContext" type="net.opengis.context.ViewContext" required="true"%>
<jsp:doBody varReader="_layer"/>
<%
com.hadrbolec.ogc.context.util.LayerUtil.add(com.hadrbolec.ogc.context.io.LayerIO.read((java.io.Read
er)jspContext.findAttribute("_layer")),(net.opengis.context.ViewContext)jspContext.findAttribute("view
Context"));
%>
```

Figure 177 Tag file: AddLayer.tag

OrderLayer

Rearranges the order of the “Layer”’s in the “LayerList” element of a given view context.

```
<context:OrderLayer layerName=”cite:NamedPlace” action=”up” viewContext=”The viewContext”/>
```

Figure 178 Example: Move layer up

The “up” action moves the “cite:NamedPlaces” layer one position upwards.

```
<context:OrderLayer layerName=”cite:NamedPlace” action=”down” viewContext=”The
viewContext”/>
```

Figure 179 Example: Move layer down

The “down” action moves the “cite:NamedPlaces” layer one position downwards.

```
<context:OrderLayer layerName=”cite:NamedPlace” action=”top” viewContext=”The viewContext”/>
```

Figure 180 Example: Move layer top

The “to” action moves the “cite:NamedPlaces” layer at the top position.

```
<context:OrderLayer layerName=”cite:NamedPlace” action=”bottom” viewContext=”The
viewContext”/>
```

Figure 181 Example: move layer bottom

The “bottom” action moves the “cite:NamedPlaces” layer at the bottom position.

```
<%@tag description = "Changes the order of the layer" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@attribute name="layerName" type="java.lang.String" required="true"%>
<%@attribute name="action" type="java.lang.String" required="true"%>
<%@attribute name="viewContext" type="net.opengis.context.ViewContext" required="true"%>
<c:choose>
  <c:when test="${action=='up'}">
    <%com.hadrbolec.ogc.context.util.OrderUtil.moveLayerUp(layerName,viewContext);%>
  </c:when>
  <c:when test="${action=='down'}">
    <%com.hadrbolec.ogc.context.util.OrderUtil.moveLayerDown(layerName,viewContext);%>
  </c:when>
  <c:when test="${action=='top'}">
    <%com.hadrbolec.ogc.context.util.OrderUtil.moveLayerTop(layerName,viewContext);%>
  </c:when>
  <c:when test="${action=='bottom'}">
    <%com.hadrbolec.ogc.context.util.OrderUtil.moveLayerBottom(layerName,viewContext);%>
  </c:when>
</c:choose>
```

Figure 182 Tag file: OrderLayer.tag

Symbolize

StyleLayer

This tag allows select a style for a layer which is contained in its StyleList element. Therefore, the current attribute of the style is set to “true”.

```
<context:StyleLayer layerName=”Name” styleName=”Name” viewContext=”ViewContext”/>
```

Figure 183 Example: select style

```
<%@tag description = "put the tag description here" pageEncoding="UTF-8" body-
content="EMPTY"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%-- The list of normal or fragment attributes can be specified here: --%>
<%@attribute name="layerName" type="java.lang.String" required="true"%>
<%@attribute name="styleName" type="java.lang.String" required="true"%>
<%@attribute name="viewContext" type="net.opengis.context.ViewContext" required="true"%>
<%com.hadrbolec.ogc.context.util.StyleUtil.setCurrent(layerName,styleName,viewContext);%>
```

Figure 184 Tag file: StyleLayer.tag

AddStyle

Adds a new “Style” to the “Layers”’s “StyleList” element. The content between the “AddStyle” tag is parsed into a “Style” element. In the case a “Style” with the same name exists it is substituted with the new one. Otherwise the new style is added after the last one. It is possible to enter any style element valid according to the WMC specification. Once the style is added or updated it is set automatically active (current attribute is set to ’true’).

```
<context:AddStyle layerName=”The layer name” viewContext="The ViewContext">
        The layer XML fragment
</context:AddStyle>
```

Figure 185 Example: add style

```
<%@tag description = "Add a new Style" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@attribute name="layerName" type="java.lang.String" required="true"%>
<%@attribute name="viewContext" type="net.opengis.context.ViewContext" required="true"%>
<jsp:doBody varReader="_style"/>
<%
com.hadrbolec.ogc.context.util.StyleUtil.add(com.hadrbolec.ogc.context.io.StyleIO.read((java.io.Reader
)jspContext.findAttribute("_style")),(net.opengis.context.ViewContext)jspContext.findAttribute("viewC
ontext"));
%>
```

Figure 186 Tag file: AddStyle.tag

Browse

ConvertPixel2Mapunit

The Convert tag is simply an internally used utility tag. It converts pixel coordinates into map units. It is used inside the move and measure tags. The first example converts a dx distance and dy distance from pixel into map units.

```
<context:ConvertPixel2Mapunit dx="400" dy="200" viewContext="The ViewContext"/>
        distanceX and distanceY variable store the distance in mapunits
```

Figure 187 Example: convert dx, dy from pixel into mapunits

The second example converts a point from pixel coordinates into map units.

```
<context:ConvertPixel2Mapunit x="400" y="200" viewContext="The ViewContext"/>
        point  variable store the coordinates of the point in mapunits
```

Figure 188 Example: convert point from pixel into map units

```
<context:ConvertPixel2Mapunit minx="200" miny="100" maxx="300" maxy="400"
viewContext="The ViewContext"/>
        boundingBox  variable store the coordinates of the point in mapunits
```

Figure 189 Example: convert bounding box from pixel into mapunits

The third and last example converts a bounding box (minx, miny, maxx, maxy) from pixel into map units.

```
<%@tag description = "Converts pixel into mapunits" pageEncoding="UTF-8"  body-
content="empty"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@attribute name="viewContext" type="net.opengis.context.ViewContext" required="true"%>
<%@attribute name="x" type="java.lang.String" required="false"%>
<%@attribute name="y" type="java.lang.String" required="false"%>
<%@attribute name="minx" type="java.lang.String" required="false"%>
<%@attribute name="miny" type="java.lang.String" required="false"%>
<%@attribute name="maxx" type="java.lang.String" required="false"%>
<%@attribute name="maxy" type="java.lang.String" required="false"%>
<%@attribute name="dx" type="java.lang.String" required="false"%>
<%@attribute name="dy" type="java.lang.String" required="false"%>
<%@variable  name-given="point" variable-class="com.hadrbolec.ogc.context.geom.Point"
scope="AT_END"%>
<%@variable  name-given="boundingBox" variable-class="net.opengis.context.BoundingBox"
scope="AT_END"%>
<%@variable  name-given="distanceX" variable-class="java.lang.Double" scope="AT_END"%>
<%@variable  name-given="distanceY" variable-class="java.lang.Double" scope="AT_END"%>
<c:choose>
   <%-- converts a point box --%>
   <c:when test="${x!=null && y!=null}">
```

```
    <c:set var="point" value="<%=com.hadrbolec.ogc.context.util.ConvertUtil.pixel2Mapunits(new
com.hadrbolec.ogc.context.geom.PointImpl(Double.parseDouble(x),Double.parseDouble(y)),viewCont
ext)%>"/>
  </c:when>
  <%-- converts a bounding box --%>
  <c:when test="${minx!=null && miny!=null && maxx!=null && maxy!=null}">
    <c:set var="boundingBox"
value="<%=com.hadrbolec.ogc.context.util.ConvertUtil.pixel2Mapunits(com.hadrbolec.ogc.context.ut
il.BoundingBoxUtil.getBoundingBox(Double.parseDouble(minx),Double.parseDouble(miny),Double.
parseDouble(maxx),Double.parseDouble(maxy)),viewContext)%>"/>
  </c:when>
  <%-- converts a dx and dy distance --%>
  <c:when test="${dx!=null && dy!=null}">
    <c:set var="distanceX"
value="<%=com.hadrbolec.ogc.context.util.ConvertUtil.dxPixel2Mapunits(Double.parseDouble(dx),vi
ewContext)%>"/>
    <c:set var="distanceY"
value="<%=com.hadrbolec.ogc.context.util.ConvertUtil.dyPixel2Mapunits(Double.parseDouble(dy),vi
ewContext)%>"/>
  </c:when>
</c:choose>
```

Figure 190 Tag file: ConvertPixelToMapunit.tag

Move

This tag moves the visible map extent according to the users input. It is necessary to convert the given coordinates from pixel values into map units (see convert p. 144).

```
<context:Move dx=”100” dy=”50”  viewContext="The ViewContext"/>
```

Figure 191 Example: move dx, dy

The first example moves the extent 100 pixels westwards and 50 pixel southwards.

```
<context:Move fromX=”100” fromY=”50”  toX=”150” toY=”20” viewContext="The ViewContext"/>
```

Figure 192 Example: move from point to point (pan)

The pan example moves the displayed map extent from the point 100, 50 to the point 150, 20.

```
<context:Move toX=”100” toY=”50”  viewContext="The ViewContext"/>
```

Figure 193 Move tag: move to point (centre at) example

The last example re-centres the extent at the point 100, 50.

```
<%@tag description = "put the tag description here" pageEncoding="UTF-8" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@taglib tagdir="/WEB-INF/tags/context" prefix="context"%>
<%@attribute name="viewContext" type="net.opengis.context.ViewContext" required="true"%>
<%@attribute name="dx" type="java.lang.String" required="false"%>
<%@attribute name="dy" type="java.lang.String" required="false"%>
<%@attribute name="fromX" type="java.lang.String" required="false"%>
```

```
<%@attribute name="fromY" type="java.lang.String" required="false"%>
<%@attribute name="toX" type="java.lang.String" required="false"%>
<%@attribute name="toY" type="java.lang.String" required="false"%>
<c:choose>
   <%-- move from point to point (pan) --%>
   <c:when test="${fromX!=null && fromY!=null && toX!=null && toY!=null}">
      <context:ConvertPixel2Mapunit x="${fromX}" y="${fromY}" viewContext="${viewContext}"/>
      <%com.hadrbolec.ogc.context.geom.Point fromPoint = new
com.hadrbolec.ogc.context.geom.PointImpl(point.getX(),point.getY());%>
      <context:ConvertPixel2Mapunit x="${toX}" y="${toY}" viewContext="${viewContext}"/>
      <%com.hadrbolec.ogc.context.geom.Point toPoint = new
com.hadrbolec.ogc.context.geom.PointImpl(point.getX(),point.getY());%>
      <%com.hadrbolec.ogc.context.util.MoveUtil.move(viewContext,fromPoint,toPoint);%>
   </c:when>
   <%-- move to point (centre at) --%>
   <c:when test="${(fromX==null || fromY==null) && (toX != null && toY != null)}">
      <context:ConvertPixel2Mapunit x="${toX}" y="${toY}" viewContext="${viewContext}"/>
      <%com.hadrbolec.ogc.context.util.MoveUtil.move(viewContext,point);%>
   </c:when>
   <%-- move dx, dy --%>
   <c:when test="${dx!=null && dy!=null}">
      <context:ConvertPixel2Mapunit dx="${dx}" dy="${dy}" viewContext="${viewContext}"/>
      <%com.hadrbolec.ogc.context.util.MoveUtil.move(viewContext,distanceX,distanceY);%>
   </c:when>
</c:choose>
```

Figure 194 Tag file: Move.tag

GetFeatureInfo

This tag file performs a WMS “GetFeatureInfo” request.

```
<context:GetFeatureInfo x="300" y="75" layerName="cite:NamedPlaces" viewContext="The
ViewContext">
request  variable store GetFeatureInfo request of the WMS service.
response  variable store GetFeatureInfo response of the WMS service.
</context:GetFeatureInfo>
```

Figure 195 GetFeatureInfo example

The “request” variable stores the WMS “GetFeatureInfo” request. The “response” variable stores the response from the service.

```
<%@tag description = "put the tag description here" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@attribute name="viewContext" type="net.opengis.context.ViewContext" rtexprvalue="true"
required="true"%>
<%@attribute name="layerName" type="java.lang.String" required="true"%>
<%@attribute name="x" type="java.lang.String" required="true"%>
<%@attribute name="y" type="java.lang.String" required="true"%>
<%@variable  name-given="response" variable-class="java.lang.String" scope="NESTED"%>
```

```
<%@variable  name-given="request" variable-class="java.lang.String" scope="NESTED"%>
<c:set var="request"
value="<%=com.hadrbolec.ogc.context.util.WMSGetRequestUtil.getGetFeatureInfoRequest(Integer.parseInt(x),Integer.parseInt(y),layerName,viewContext)%>"/>
<c:set var="response"><c:import url="${request}"/></c:set>
<jsp:doBody/>
```

Figure 196 Tag file: GetFeatureInfo.tag

List / report

GetFeature

The GetFeature tag creates, based on the given LayerFeatureConstraint element, a WFS "GetFeature" request. The request is performed and the response is stored.

```
<context:GetFeature layerName="The layer name" maxFeatures="maximal number of features"
viewContext="The view context">
    ${features.request} Stores the GetFeature request
    ${features.response} Stores the GetFeature response
</context:GetFeature>
```

Figure 197 Example: WFS GetFeature request

The "features" variable stores the WFS "GetFeature" request and response.

```
<%@tag description = "put the tag description here" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@taglib tagdir="/WEB-INF/tags/io" prefix="io"%>
<%@attribute name="viewContext" type="net.opengis.context.ViewContext" rtexprvalue="true"
required="true"%>
<%@attribute name="layerName" type="java.lang.String" required="true"%>
<%@attribute name="maxFeatures" type="java.lang.String" required="true"%>
<%@variable  name-given="features" variable-class="com.hadrbolec.ogc.context.dto.FeatureDTO"
scope="NESTED"%>
<c:set var="features"
value="<%=com.hadrbolec.ogc.context.render.FeatureRendererHelper.getFeatures(layerName,Integer.parseInt(maxFeatures),viewContext)%>"/>
<jsp:doBody/>
```

Figure 198 Tag file: GetFeature.tag

Data manipulation and Analysis functions

Scale

This tag scales the displayed map extent.

```
<context:Scale factor="The factor" ViewContext="The view context"/>
```

Figure 199 Scale factor example

The first example scales the displayed bounding box with the provided factor.

```
<context: Scale x=”x coordinate” y=”y coordinate” factor=”The factor” viewContext=" The view context"/>
```

Figure 200 Scale centre factor

In this example the map is re-centred at the given point and afterwards scaled according to the given factor.

```
<context:Scale minx="min x" miny="min y" maxx="max x" maxy="max y" viewContext=" The view context"/>
```

Figure 201 Scale to bounding box example

This example scales to the given bounding box.

```
<%@tag description = "put the tag description here" pageEncoding="UTF-8" body-content="EMPTY" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@taglib tagdir="/WEB-INF/tags/context" prefix="context"%>
<%@attribute name="viewContext" type="net.opengis.context.ViewContext" required="true"%>
<%@attribute name="factor" type="java.lang.String" required="false"%>
<%@attribute name="x" type="java.lang.String" required="false"%>
<%@attribute name="y" type="java.lang.String" required="false"%>
<%@attribute name="minx" type="java.lang.String" required="false"%>
<%@attribute name="miny" type="java.lang.String" required="false"%>
<%@attribute name="maxx" type="java.lang.String" required="false"%>
<%@attribute name="maxy" type="java.lang.String" required="false"%>
<%@attribute name="units" type="java.lang.String" required="false"%>
<c:choose>
  <%-- zoom factor --%>
  <c:when test="${x==null && y==null && factor!=null}">
    <%com.hadrbolec.ogc.context.util.ScaleUtil.scale(viewContext,Double.parseDouble(factor));%>
  </c:when>
  <%-- zoom center factor --%>
  <c:when test="${x!=null && y!=null && factor!=null}">
    <context:ConvertPixel2Mapunit x="${x}" y="${y}" viewContext="${viewContext}"/>
    <%com.hadrbolec.ogc.context.util.ScaleUtil.scale(viewContext,Double.parseDouble(factor));%>
  </c:when>
  <%-- rescale to bounding box (zoom rectangle) --%>
  <c:when test="${minx!=null && miny!=null && maxx!=null && maxy!=null}">
  <c:choose>
    <c:when test="${units!=null && units=='mapunits'}">
      <%
      net.opengis.context.BoundingBox boundingBox =
com.hadrbolec.ogc.context.util.BoundingBoxUtil.getBoundingBox(Double.parseDouble(minx),
Double.parseDouble(miny),Double.parseDouble(maxx),Double.parseDouble(maxy));
      com.hadrbolec.ogc.context.util.ScaleUtil.scale(viewContext,boundingBox);
      %>
    </c:when>
    <c:otherwise>
      <context:ConvertPixel2Mapunit minx="${minx}" miny="${miny}" maxx="${maxx}"
```

```
maxy="${maxy}" viewContext="${viewContext}"/>
        <%com.hadrbolec.ogc.context.util.ScaleUtil.scale(viewContext,boundingBox);%>
      </c:otherwise>
   </c:choose>
   </c:when>
</c:choose>
```

Figure 202 Scale.tag

The tag file examines the input parameters provided and determines the necessary action(s) according to them.

Measure

The measure tag allows to measure a distance. The given coordinates are converted from pixel into map units. The given x- and y-distance difference in pixel is converted into map units. Afterwards the distance is calculated. The separate conversion in x and y direction is necessary, because the displayed map can be scaled differently in x and y direction.

```
<context:Measure dx="200" dy="100" viewContext="The ViewContext">
        Distance as distance variable
</context:Measure>
```

Figure 203 Example: measure

```
<%@tag description = "put the tag description here" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@taglib tagdir="/WEB-INF/tags/context" prefix="context"%>
<%@attribute name="viewContext" type="net.opengis.context.ViewContext" required="true"%>
<%@attribute name="dx" type="java.lang.String" required="true"%>
<%@attribute name="dy" type="java.lang.String" required="true"%>
<%@variable  name-given="distance" variable-class="java.lang.Double" scope="NESTES"%>
<context:ConvertPixel2Mapunit dx="${dx}" dy="${dy}" viewContext="${viewContext}"/>
<c:set var="distance"><%=Math.hypot(distanceX.doubleValue(),distanceY.doubleValue())%></c:set>
<jsp:doBody/>
```

Figure 204 Measure.tag

Spatial Analysis

Graphical overplot

GetMap

This tag generates a WMS “GetMap” request for all visible layers in a “ViewContext” document. It graphically overplots the layers and displays the map.

```
<context:GetMap viewContext="The ViewContext"/>
```

Figure 205 Example: GetMap

The example displays a map of a given view context.

```
<%@tag description = "put the tag description here" pageEncoding="UTF-8" body-content="empty"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@taglib tagdir="/WEB-INF/tags/context" prefix="context"%>
<%@attribute name="viewContext" type="net.opengis.context.ViewContext" rtexprvalue="true"
required="true"%>
<context:DisplayMap
    map="<%=com.hadrbolec.ogc.context.render.RenderFactory.renderMap(viewContext)%>"/>
```

Figure 206 Tag file: GetMap.tag

The map is visualised with the help of the DisplayMap tag. The DisplayMap tag graphically overplots all layers which have a hidden attribute with the value false.

```
<%@tag description = "put the tag description here" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@attribute name="map" type="com.hadrbolec.ogc.context.dto.Map" required="true"%>
<div
style="position:relative;width:${map.window.width+6}px;height:${map.window.height+6}px;border-
style:outset;border-width:3px">
  <c:forEach var="item" items="${map.images}">
    <img style="position:absolute;top:0px;left:0px" src="${item.URI}"
width="${map.window.width}" width="${map.window.height}"/>
  </c:forEach>
</div>
```

Figure 207 DisplayMap.tag

Utility classes

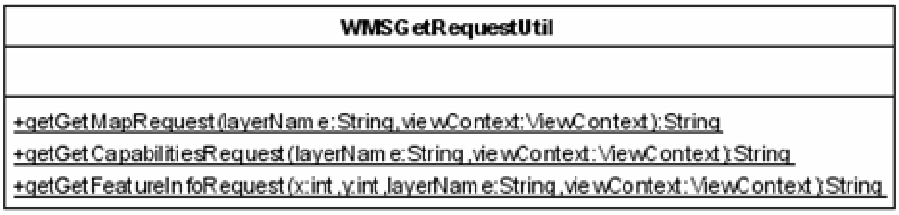


Figure 208 WMSGetRequestUtil

This class creates for a given ViewContext a WMS “GetMap”, “GetCapabilities” and / or “GetFeatureInfo” request.

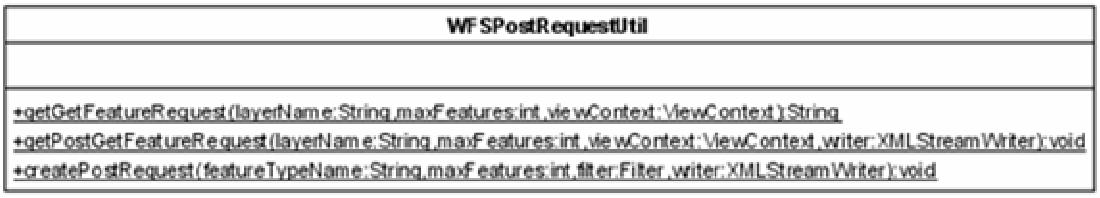


Figure 209 WFSPostRequestUtil

The “WFSPostRequestUtil” class performs a WFS “GetFeature” request with the post method.

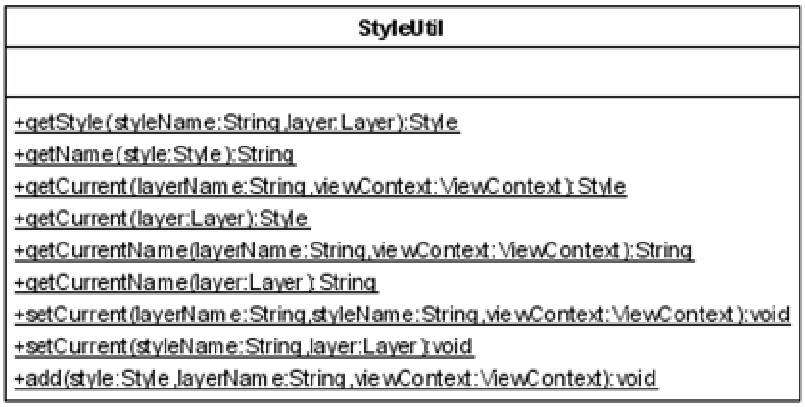


Figure 210 StyleUtil

This utility gets, sets the current style and / or adds a new one.

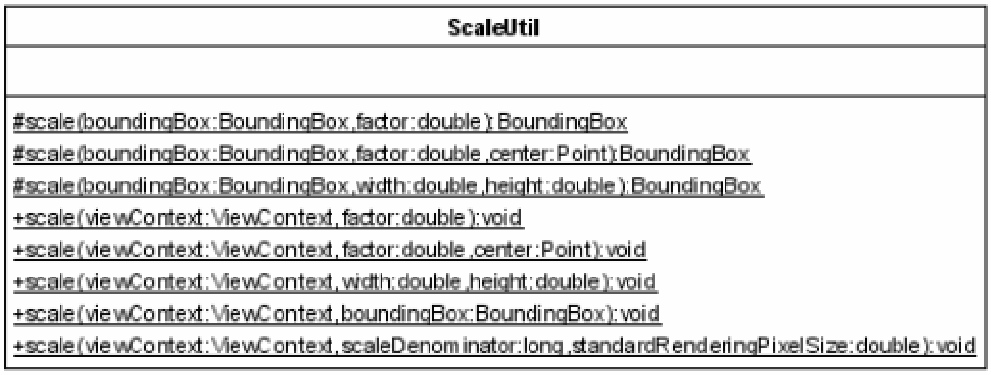


Figure 211 ScaleUtil

The "ScaleUtil" performs all necessary scaling functions.

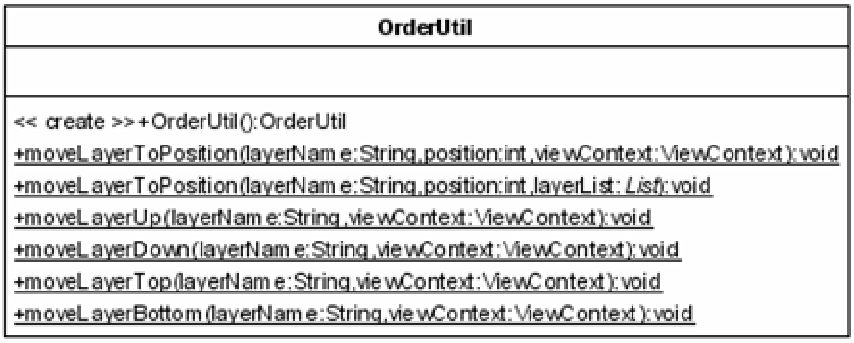


Figure 212 OrderUtil

This utility allows to change the order of the layers in a ViewContext document.

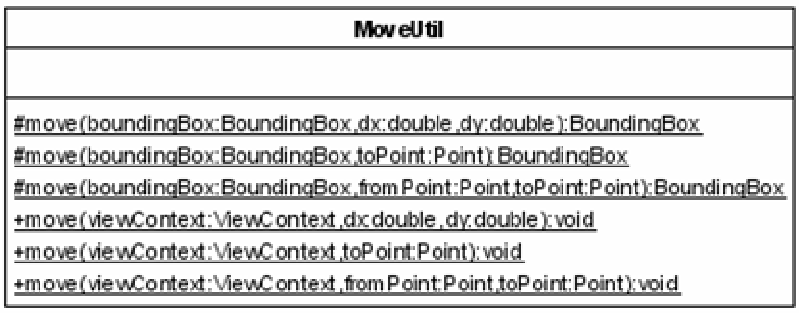


Figure 213 MoveUtil

The “MoveUtil” performs all necessary move functions in a ViewContext document.

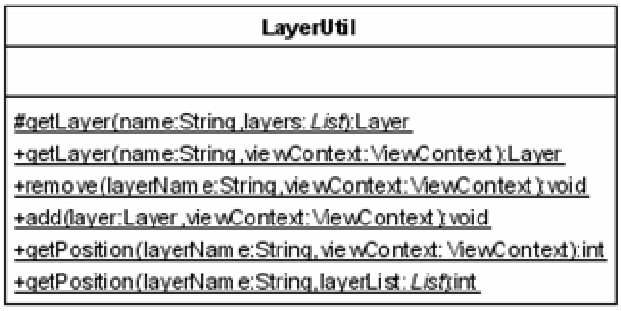


Figure 214 LayerUtil

With this utility class it is possible to add, remove and get a layer of a ViewContext document. Additionally, the position of a layer can be determined.

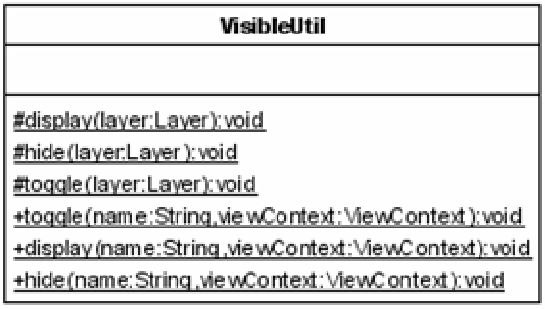


Figure 215 VisibleUtil

The visibility of a layer is changed (display, hide, toggle) with the help of this utility.

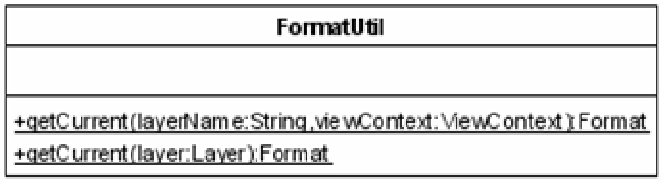


Figure 216 FormatUtil

The “FormatUtil” allows to set and / or get the current format of a layer in a ViewContext document.

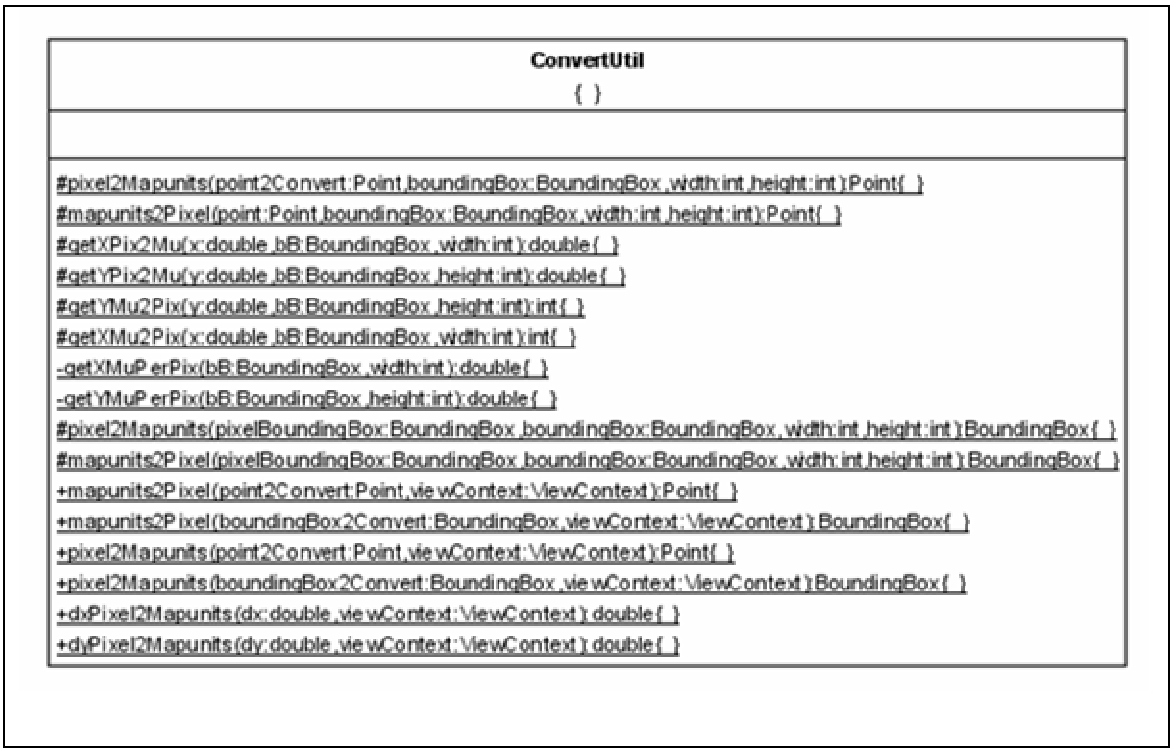


Figure 217 ConvertUtil

The “ConvertUtil” contains all functionalities needed to convert distances, points and bounding boxes from pixel into map units.

ViewContext documents

RemoteWFS_States.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<ViewContext xmlns="http://www.opengis.net/context" xmlns:sld="http://www.opengis.net/sld"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.0.0" id="U1051_States">
  <General>
    <Window width="400" height="200"/>
    <BoundingBox SRS="EPSG:4326" minx="-126" miny="18" maxx="-66" maxy="57"/>
    <Title>Layer order 1</Title>
  </General>
  <LayerList>
    <Layer queryable="true" hidden="false">
      <Server service="OGC:WMS" version="1.1.1">
        <OnlineResource xlink:type="simple" xlink:href="http://127.0.0.1:8084/deegreewms/wms?"/>
      </Server>
      <Name>States</Name>
      <Title>States</Title>
      <SRS>EPSG:4326</SRS>
      <FormatList>
        <Format current="true">image/gif</Format>
      </FormatList>
      <StyleList>
        <Style current="true">
          <SLD>
            <Name>Original</Name>
            <OnlineResource xlink:type="simple"
              xlink:href="http://127.0.0.1:8084/deegreewms/debug/sld/States_SLD.xml"/>
          </SLD>
        </Style>
      </StyleList>
    </Layer>
  </LayerList>
</ViewContext>
```

Figure 218 "ViewContext" document: RemoteWFS_States.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<StyledLayerDescriptor xmlns="http://www.opengis.net/sld" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:gml="http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.0.0">
  <UserLayer>
    <RemoteOWS>
      <Service>WFS</Service>
      <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
        xlink:href="http://127.0.0.1:8084/geoserver/wfs?"/>
```

```
      </RemoteOWS>
      <LayerFeatureConstraints>
        <FeatureTypeConstraint>
          <FeatureTypeName>topp:states</FeatureTypeName>
        </FeatureTypeConstraint>
      </LayerFeatureConstraints>
      <UserStyle>
        <Name>topp:states</Name>
        <Title>topp:states</Title>
        <FeatureTypeStyle>
          <Rule>
            <PolygonSymbolizer>
              <Geometry>
                <ogc:PropertyName>topp:the_geom</ogc:PropertyName>
              </Geometry>
              <Stroke>
                <CssParameter name="stroke">#0000FF</CssParameter>
                <CssParameter name="stroke-width">2</CssParameter>
              </Stroke>
            </PolygonSymbolizer>
          </Rule>
        </FeatureTypeStyle>
      </UserStyle>
    </UserLayer>
</StyledLayerDescriptor>
```

Figure 219 StyledLayerDescriptor file: States_SLD.xml

This file is the online referenced StyledLayerDescriptor file for the “ViewContext” shown in Figure 218 “ViewContext” document: .

WMS_cite_WMC.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<ViewContext xmlns="http://www.opengis.net/context" xmlns:sld="http://www.opengis.net/sld"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.0.0" id="U1051_Cite">
  <General>
    <Window width="400" height="200"/>
    <BoundingBox SRS="EPSG:4326" minx="-0.005" miny="-0.005" maxx="0.005" maxy="0.005"/>
    <Title>UniGIS MSc 2003 master thesis</Title>
    <KeywordList>
      <Keyword>Master thesis</Keyword>
      <Keyword>U1051</Keyword>
    </KeywordList>
    <Abstract>Master thesis MSc UniGIS MSc 2003. This view context document stores the examples
used within the master thesis.</Abstract>
    <LogoURL format="image/gif" width="126" height="80">
      <OnlineResource xlink:type="simple"
xlink:href="http://127.0.0.1:8084/deegreewms/images/myLogo.gif"/>
    </LogoURL>
    <DescriptionURL>
      <OnlineResource xlink:type="simple"
xlink:href="http://127.0.0.1:8084/description/Description.html"/>
    </DescriptionURL>
    <ContactInformation>
      <ContactPersonPrimary>
        <ContactPerson>Michael Hadrbolec</ContactPerson>
        <ContactOrganization>Agency</ContactOrganization>
      </ContactPersonPrimary>
      <ContactPosition>Developer</ContactPosition>
      <ContactAddress>
        <Address>Elmsstreet 13</Address>
        <City>Vienna</City>
        <StateOrProvince>Vienna</StateOrProvince>
        <PostCode>1090</PostCode>
        <Country>Austria</Country>
      </ContactAddress>
      <ContactVoiceTelephone>0815</ContactVoiceTelephone>
      <ContactFacsimileTelephone>0915</ContactFacsimileTelephone>
      <ContactElectronicMailAddress>test@test.net</ContactElectronicMailAddress>
    </ContactInformation>
  </General>
  <LayerList>
    <!-- Begin: cite BasicPolygons layer -->
    <Layer queryable="true" hidden="true">
    <Server service="OGC:WMS" version="1.1.1">
```

```
        <OnlineResource xlink:type="simple" xlink:href="http://127.0.0.1:8084/deegreewms/wms?"/>
      </Server>
      <Name>cite:BasicPolygons</Name>
      <Title>BASIC Polygons</Title>
      <SRS>EPSG:4326</SRS>
      <FormatList>
        <Format>image/png</Format>
        <Format>image/jpeg</Format>
        <Format>image/tif</Format>
        <Format>image/svg+xml</Format>
        <Format>image/bmp</Format>
        <Format current="true">image/gif</Format>
        <Format>image/jpg</Format>
      </FormatList>
      <StyleList>
        <Style current="true">
          <Name>default:cite:BasicPolygons</Name>
          <Title>default:cite:BasicPolygons</Title>
          <LegendURL width="20" height="20">
            <OnlineResource xlink:type="simple"
xlink:href="http://127.0.0.1:8084/deegreewms/legend/default_cite_BasicPolygons.png"/>
          </LegendURL>
        </Style>
        <Style>
          <SLD>
            <Name>online:styledLayeDescriptorFile:cite:Descriptor:BasicPolygons</Name>
            <Title>Online basic polygon style</Title>
            <OnlineResource xlink:type="simple"
xlink:href="http://127.0.0.1:8084/deegreewms/BasicPolygons.sld"/>
          </SLD>
        </Style>
        <Style>
        <SLD>
          <Name>inline:styledLayeDescriptor:cite:Descriptor:BasicPolygons</Name>
          <Title>SLD basic polygon style</Title>
          <sld:StyledLayerDescriptor version="1.0.0">
            <sld:NamedLayer>
              <sld:Name>cite:BasicPolygons</sld:Name>
              <sld:UserStyle>
                <sld:Name>default:cite:BasicPolygons</sld:Name>
                <sld:Title>Userdefined</sld:Title>
                <sld:FeatureTypeStyle>
                  <sld:Name>default:cite:BasicPolygons</sld:Name>
                  <sld:Rule>
                    <sld:Name>cite:BasicPolygons</sld:Name>
                    <sld:PolygonSymbolizer>
                      <sld:Geometry>
                        <ogc:PropertyName>GEOM</ogc:PropertyName>
```

```
                        </sld:Geometry>
                        <sld:Stroke>
                           <sld:CssParameter name="stroke">#000000</sld:CssParameter>
                           <sld:CssParameter name="stroke-width">3</sld:CssParameter>
                        </sld:Stroke>
                     </sld:PolygonSymbolizer>
                  </sld:Rule>
               </sld:FeatureTypeStyle>
            </sld:UserStyle>
         </sld:NamedLayer>
      </sld:StyledLayerDescriptor>
   </SLD>
   </Style>
   <Style>
   <SLD>
      <Name>inline:featureTypeStyle:cite:Descriptor:BasicPolygons</Name>
      <Title>inline feature type style basic polygon style</Title>
      <sld:FeatureTypeStyle>
         <sld:Name>default:cite:BasicPolygons</sld:Name>
         <sld:Rule>
            <sld:Name>cite:BasicPolygons</sld:Name>
            <sld:PolygonSymbolizer>
               <sld:Geometry>
                  <ogc:PropertyName>GEOM</ogc:PropertyName>
               </sld:Geometry>
               <sld:Stroke>
                  <sld:CssParameter name="stroke">#FF0000</sld:CssParameter>
                  <sld:CssParameter name="stroke-width">3</sld:CssParameter>
               </sld:Stroke>
            </sld:PolygonSymbolizer>
         </sld:Rule>
      </sld:FeatureTypeStyle>
   </SLD>
   </Style>
</StyleList>
</Layer>
<!--End:: cite BasicPolygons layer -->
<!-- Begin: cite:Bridges layer -->
<Layer queryable="true" hidden="false">
   <Server service="OGC:WMS" version="1.1.1">
      <OnlineResource xlink:type="simple" xlink:href="http://127.0.0.1:8084/deegreewms/wms?"/>
   </Server>
   <Name>cite:Bridges</Name>
   <Title>CITE BRIDGES</Title>
   <SRS>EPSG:4326</SRS>
   <FormatList>
      <Format>image/png</Format>
      <Format>image/jpeg</Format>
```

```
            <Format>image/tif</Format>
            <Format>image/svg+xml</Format>
            <Format>image/bmp</Format>
            <Format current="true">image/gif</Format>
            <Format>image/jpg</Format>
          </FormatList>
          <StyleList>
            <Style current="true">
              <Name>default:cite:Buildings</Name>
              <Title>CITE BUILDINGS</Title>
              <LegendURL width="20" height="20" format="image/png">
                <OnlineResource xlink:type="simple"
xlink:href="http://127.0.0.1:8084/deegreewms/legend/default_cite_Bridges.png"/>
              </LegendURL>
            </Style>
          </StyleList>
        </Layer>
        <!-- End: cite:Bridges layer -->
        <!-- Begin: cite:BuildingCenters layer -->
        <Layer queryable="true" hidden="false">
          <Server service="OGC:WMS" version="1.1.1">
            <OnlineResource xlink:type="simple" xlink:href="http://127.0.0.1:8084/deegreewms/wms?"/>
          </Server>
          <Name>cite:BuildingCenters</Name>
          <Title>CITE BUILDINGCENTERS</Title>
          <SRS>EPSG:4326</SRS>
          <FormatList>
            <Format>image/png</Format>
            <Format>image/jpeg</Format>
            <Format>image/tif</Format>
            <Format>image/svg+xml</Format>
            <Format>image/bmp</Format>
            <Format current="true">image/gif</Format>
            <Format>image/jpg</Format>
          </FormatList>
          <StyleList>
            <Style current="true">
              <Name>default:cite:BuildingCenters</Name>
              <Title>CITE BUILDINGCENTERS</Title>
              <LegendURL width="20" height="20" format="image/png">
                <OnlineResource xlink:type="simple"
xlink:href="http://127.0.0.1:8084/deegreewms/legend/default_cite_BuildingCenters.png"/>
              </LegendURL>
            </Style>
          </StyleList>
        </Layer>
        <!-- End: cite:BuildingCenters layer -->
        <!-- Begin: The cite buildings layer -->
```

```
      <Layer queryable="true" hidden="false">
        <Server service="OGC:WMS" version="1.1.1">
          <OnlineResource xlink:type="simple" xlink:href="http://127.0.0.1:8084/deegreewms/wms?"/>
        </Server>
        <Name>cite:Buildings</Name>
        <Title>BUILDINGS</Title>
        <SRS>EPSG:4326</SRS>
        <FormatList>
          <Format>image/png</Format>
          <Format>image/jpeg</Format>
          <Format>image/tif</Format>
          <Format>image/svg+xml</Format>
          <Format>image/bmp</Format>
          <Format current="true">image/gif</Format>
          <Format>image/jpg</Format>
        </FormatList>
        <StyleList>
          <Style current="true">
            <Name>default:cite:Buildings</Name>
            <Title>CITE BUILDINGS</Title>
            <LegendURL width="20" height="20" format="image/png">
              <OnlineResource xlink:type="simple"
xlink:href="http://127.0.0.1:8084/deegreewms/legend/default_cite_Buildings.png"/>
            </LegendURL>
          </Style>
        </StyleList>
      </Layer>
      <!-- End: The cite building layer -->
      <!-- Begin: The DividedRoutes layer -->
      <Layer queryable="true" hidden="false">
        <Server service="OGC:WMS" version="1.1.1">
          <OnlineResource xlink:type="simple" xlink:href="http://127.0.0.1:8084/deegreewms/wms?"/>
        </Server>
        <Name>cite:DividedRoutes</Name>
        <Title>CITE DIVIDED ROUTES</Title>
        <SRS>EPSG:4326</SRS>
        <FormatList>
          <Format>image/png</Format>
          <Format>image/jpeg</Format>
          <Format>image/tif</Format>
          <Format>image/svg+xml</Format>
          <Format>image/bmp</Format>
          <Format current="true">image/gif</Format>
          <Format>image/jpg</Format>
        </FormatList>
        <StyleList>
          <Style current="true">
            <Name>default:cite:DividedRoutes</Name>
```

```
            <Title>default:cite:DividedRoutes</Title>
            <LegendURL width="20" height="20" format="image/png">
              <OnlineResource xlink:type="simple"
xlink:href="http://127.0.0.1:8084/deegreewms/legend/default_cite_DividedRoutes.png"/>
            </LegendURL>
          </Style>
        </StyleList>
      </Layer>
      <!-- End: The DividedRoutes layer -->
      <!-- Begin: The Forests layer -->
      <Layer queryable="true" hidden="false">
        <Server service="OGC:WMS" version="1.1.1">
          <OnlineResource xlink:type="simple" xlink:href="http://127.0.0.1:8084/deegreewms/wms?"/>
        </Server>
        <Name>cite:Forests</Name>
        <Title>CITE FORESTS</Title>
        <SRS>EPSG:4326</SRS>
        <FormatList>
          <Format>image/png</Format>
          <Format>image/jpeg</Format>
          <Format>image/tif</Format>
          <Format>image/svg+xml</Format>
          <Format>image/bmp</Format>
          <Format current="true">image/gif</Format>
          <Format>image/jpg</Format>
        </FormatList>
        <StyleList>
          <Style current="true">
            <Name>default:cite:Forests</Name>
            <Title>default:cite:Forests</Title>
            <LegendURL width="20" height="20" format="image/png">
              <OnlineResource xlink:type="simple"
xlink:href="http://127.0.0.1:8084/deegreewms/legend/default_cite_Forests.png"/>
            </LegendURL>
          </Style>
        </StyleList>
      </Layer>
      <!-- End: The Forests layer -->
      <!-- Begin: The Lakes layer -->
      <Layer queryable="true" hidden="false">
        <Server service="OGC:WMS" version="1.1.1">
          <OnlineResource xlink:type="simple" xlink:href="http://127.0.0.1:8084/deegreewms/wms?"/>
        </Server>
        <Name>cite:Lakes</Name>
        <Title>CITE LAKES</Title>
        <SRS>EPSG:4326</SRS>
        <FormatList>
          <Format>image/png</Format>
```

```
            <Format>image/jpeg</Format>
            <Format>image/tif</Format>
            <Format>image/svg+xml</Format>
            <Format>image/bmp</Format>
            <Format current="true">image/gif</Format>
            <Format>image/jpg</Format>
          </FormatList>
          <StyleList>
            <Style>
              <Name>BlueFill</Name>
              <Title>BlueFill</Title>
              <LegendURL width="25" height="25" format="image/png">
                <OnlineResource xlink:type="simple"
xlink:href="http://127.0.0.1:8084/deegreewms/legend/BlueFill.png"/>
              </LegendURL>
            </Style>
            <Style current="true">
              <Name>default:cite:Lakes</Name>
              <Title>default:cite:Lakes</Title>
              <LegendURL width="20" height="20" format="image/png">
                <OnlineResource xlink:type="simple"
xlink:href="http://127.0.0.1:8084/deegreewms/legend/default_cite_Lakes.png"/>
              </LegendURL>
            </Style>
          </StyleList>
        </Layer>
        <!-- End: The Lakes layer -->
        <!-- Begin: The MapNeatline layer -->
        <Layer queryable="false" hidden="false">
          <Server service="OGC:WMS" version="1.1.1">
            <OnlineResource xlink:type="simple" xlink:href="http://127.0.0.1:8084/deegreewms/wms?"/>
          </Server>
          <Name>cite:MapNeatline</Name>
          <Title>CITE MAPNEATLINE</Title>
          <SRS>EPSG:4326</SRS>
          <FormatList>
            <Format>image/png</Format>
            <Format>image/jpeg</Format>
            <Format>image/tif</Format>
            <Format>image/svg+xml</Format>
            <Format>image/bmp</Format>
            <Format current="true">image/gif</Format>
            <Format>image/jpg</Format>
          </FormatList>
          <StyleList>
            <Style>
              <Name>GreenLine</Name>
              <Title>GreenLine</Title>
```

```
            <LegendURL width="25" height="25" format="image/png">
              <OnlineResource xlink:type="simple"
xlink:href="http://127.0.0.1:8084/deegreewms/legend/GreenLine.png"/>
            </LegendURL>
          </Style>
          <Style current="true">
            <Name>default:cite:MapNeatline</Name>
            <Title>default:cite:MapNeatline</Title>
            <LegendURL width="20" height="20" format="image/png">
              <OnlineResource xlink:type="simple"
xlink:href="http://127.0.0.1:8084/deegreewms/legend/default_cite_MapNeatline.png"/>
            </LegendURL>
          </Style>
        </StyleList>
      </Layer>
      <!-- End: The MapNeatline layer -->
      <!-- Begin: The NamedPlaces layer -->
      <Layer queryable="true" hidden="false">
        <Server service="OGC:WMS" version="1.1.1">
          <OnlineResource xlink:type="simple" xlink:href="http://127.0.0.1:8084/deegreewms/wms?"/>
        </Server>
        <Name>cite:NamedPlaces</Name>
        <Title>CITE NAMEDPLACES</Title>
        <SRS>EPSG:4326</SRS>
        <FormatList>
          <Format>image/png</Format>
          <Format>image/jpeg</Format>
          <Format>image/tif</Format>
          <Format>image/svg+xml</Format>
          <Format>image/bmp</Format>
          <Format current="true">image/gif</Format>
          <Format>image/jpg</Format>
        </FormatList>
        <StyleList>
          <Style current="true">
            <Name>default:cite:NamedPlaces</Name>
            <Title>default:cite:NamedPlaces</Title>
            <LegendURL width="20" height="20" format="image/png">
              <OnlineResource xlink:type="simple"
xlink:href="http://127.0.0.1:8084/deegreewms/legend/default_cite_NamedPlaces.png"/>
            </LegendURL>
          </Style>
        </StyleList>
      </Layer>
      <!-- End: The namedPlaces layer -->
      <!-- Begin: The ponds layer -->
      <Layer queryable="true" hidden="false">
        <Server service="OGC:WMS" version="1.1.1">
```

```
            <OnlineResource xlink:type="simple" xlink:href="http://127.0.0.1:8084/deegreewms/wms?"/>
          </Server>
          <Name>cite:Ponds</Name>
          <Title>CITE PONDS</Title>
          <SRS>EPSG:4326</SRS>
          <FormatList>
            <Format>image/png</Format>
            <Format>image/jpeg</Format>
            <Format>image/tif</Format>
            <Format>image/svg+xml</Format>
            <Format>image/bmp</Format>
            <Format current="true">image/gif</Format>
            <Format>image/jpg</Format>
          </FormatList>
          <StyleList>
            <Style current="true">
              <Name>default:cite:Ponds</Name>
              <Title>default:cite:Ponds</Title>
              <LegendURL width="20" height="20" format="image/png">
                <OnlineResource xlink:type="simple"
xlink:href="http://127.0.0.1:8084/deegreewms/legend/default_cite_Ponds.png"/>
              </LegendURL>
            </Style>
            <Style>
              <Name>RedFill</Name>
              <Title>RedFill</Title>
              <LegendURL width="25" height="25" format="image/png">
                <OnlineResource xlink:type="simple"
xlink:href="http://127.0.0.1:8084/deegreewms/legend/RedFill.png"/>
              </LegendURL>
            </Style>
          </StyleList>
        </Layer>
        <!-- End: The ponds layer -->
        <!-- Begin: The RoadSegments layer -->
        <Layer queryable="true" hidden="false">
          <Server service="OGC:WMS" version="1.1.1">
            <OnlineResource xlink:type="simple" xlink:href="http://127.0.0.1:8084/deegreewms/wms?"/>
          </Server>
          <Name>cite:RoadSegments</Name>
          <Title>CITE ROADSEGMENTS</Title>
          <SRS>EPSG:4326</SRS>
          <FormatList>
            <Format>image/png</Format>
            <Format>image/jpeg</Format>
            <Format>image/tif</Format>
            <Format>image/svg+xml</Format>
            <Format>image/bmp</Format>
```

```
            <Format current="true">image/gif</Format>
            <Format>image/jpg</Format>
          </FormatList>
          <StyleList>
            <Style current="true">
              <Name>default:cite:RoadSegments</Name>
              <Title>default:cite:RoadSegments</Title>
              <LegendURL width="20" height="20" format="image/png">
                <OnlineResource xlink:type="simple"
xlink:href="http://127.0.0.1:8084/deegreewms/legend/default_cite_RoadSegments.png"/>
              </LegendURL>
            </Style>
          </StyleList>
        </Layer>
        <!-- End: The RoadSegments layer -->
        <!-- Begin: The Streams layer -->
        <Layer queryable="false" hidden="false">
          <Server service="OGC:WMS" version="1.1.1">
            <OnlineResource xlink:type="simple" xlink:href="http://127.0.0.1:8084/deegreewms/wms?"/>
          </Server>
          <Name>cite:Streams</Name>
          <Title>CITE STREAMS</Title>
          <SRS>EPSG:4326</SRS>
          <FormatList>
            <Format>image/png</Format>
            <Format>image/jpeg</Format>
            <Format>image/tif</Format>
            <Format>image/svg+xml</Format>
            <Format>image/bmp</Format>
            <Format current="true">image/gif</Format>
            <Format>image/jpg</Format>
          </FormatList>
          <StyleList>
            <Style>
              <Name>GreenLine</Name>
              <Title>GreenLine</Title>
              <LegendURL width="25" height="25" format="image/png">
                <OnlineResource xlink:type="simple"
xlink:href="http://127.0.0.1:8084/deegreewms/legend/GreenLine.png"/>
              </LegendURL>
            </Style>
            <Style current="true">
              <Name>default:cite:Streams</Name>
              <Title>default:cite:Streams</Title>
              <LegendURL width="20" height="20" format="image/png">
                <OnlineResource xlink:type="simple"
xlink:href="http://127.0.0.1:8084/deegreewms/legend/default_cite_Streams.png"/>
              </LegendURL>
```

```
            </Style>
          </StyleList>
        </Layer>
        <!-- End: The Streams layer -->
    </LayerList>
</ViewContext>
```

Figure 220 “ViewContext” document: WMS_cite_WMC.xml