Master Thesis

submitted within the UNIGIS MSc programme
at the Centre for GeoInformatics (Z_GIS)
Salzburg University

Towards a 4D WebGIS using harmonised datasets

Examined on an New Zealand example

by

Dipl.-Ing. Alexander Kmoch
472849

A thesis submitted in partial fulfilment of the requirements of
the degree of
Master of Science (Geographical Information Science & Systems) – MSc (GISc)

Advisor:

Dr. Hermann Klug

Salzburg, Austria, 02.07.2012

Abstract

New Zealand's groundwater resources are extremely valuable but poorly understood. There are 16 regional and unitary councils, which are responsible for the freshwater management within their region. Future consumptive use of groundwater will increase because surface water is already fully allocated in many catchments. Groundwater also provides base flow to rivers, streams, springs and lakes, which are vital to the tourism industry and central to the cultural and recreational values of New Zealanders. Therefore the SMART project was started in New Zealand. The overall aim of this programme is to assemble and validate a suite of highly innovative techniques that can be applied to map and characterise New Zealand's aquifers. The research outcome will support research and management policies for groundwater resources and decision making.

This thesis takes part in the contributing efforts of the Z_GIS (Austria) in the context of data synthesis/harmonisation and visualisation. Existing groundwater related datasets of the regional councils and research institutes in their respective formats have been gathered and analysed. An architecture for distributed datasets and a common database scheme has to be refined, which is capable of integrating the data from different sources. The spatial data infrastructure should provide means of storing, accessing and rendering hydro(geo)logical data in 3D with an additional time dimension and support future downstream rendering of 3D/4D models in an open, easy to access browser-based web application. Furthermore an interface for connection with sensor networks is considered.

Keywords: New Zealand, WebGIS, sensor web, hydrology, geology, 3d, time-series

Table of Contents

List of Figures

List of Tables

List of Abbreviations

SOS	Sensor Observation Service
SWE	Sensor Web Enablement
NGMP	National Groundwater Monitoring Programme
GGW	GNS Science Geothermal and Groundwater Database
GPU	Graphics Processing Unit
WMS	Web Mapping Service
WFS	Web Feature Service
WCS	Web Coverage Service
CSW	Catalogue Service for Web
OGC	Open Geospatial Consortium
X3D	Extensible 3D
CSS	Cascading Style Sheets
SLD	Styled Layer Definition
XML	Extensible Markup Language
GML	Geographic Markup Language
O&M	Observations and Measurements
HTML	Hypertext Markup Language
HUD	Heads Up Display
UML	Unified Modelling Language
GNS	Institute of Geological and Nuclear Sciences
SDI	Spatial Data Infrastructure
OGC	Open Geospatial Consortium
XSL	eXtensible Stylesheet Language
XSLT	eXtensible Stylesheet Language Transformation
SQL	Structured Query Language
GWML	Groundwater Markup Language

Acknowledgements

I would like to thank Dr. Hermann Klug for his ever present guidance and support. Also his trust and belief in me throughout the project and his exemplary function as role model inspired me. I would like to thank the whole UNIGIS Salzburg team for the excellent study courses and their valuable feedback. Especially I would like to name Professor Josef Strobl, Dr. Adrijana Car, Dr. Shahnawaz, Anna Karnassioti, Dr. Gudrun Wallentin and Simone Duerr. Furthermore I would like to thank the SMART project team and the GNS colleagues in New Zealand, who welcomed me and provided a fantastic work environment. And I would like to thank my family and friends for being patient and understanding about my lack of regular communication. I am very happy to have you all.

1 Introduction

New Zealand is divided into 16 regions that are governed by the regional councils. Natural resources are managed by the regional councils. New Zealand's surface freshwater sources are already widely allocated. As freshwater demands grow, because of extensive irrigation use in agriculture for example, groundwater allocation is getting more important. But New Zealand's groundwater resources are not well understood (White, 2006). The SMART project (Klug et al., 2011; SMART, 2012) aims to characterise New Zealand's aquifers to support regional councils and other stakeholders in policy making and sustainable groundwater management. Within the project, which is funded by the Ministry of Science and Innovation New Zealand (MSI, 2012), generated datasets as well as already existing hydrological and geological datasets of participating regional councils and research institutes are supposed to be made available in a one-stop-shop web portal. To follow this approach datasets need to be technically and semantically harmonised (Atkinson et al., 2012). The web portal should provide services to discover, view and serve access to these datasets to provide a central information hub for New Zealand's groundwater data. Furthermore the portal should provide means to explore the datasets in 2D and 3D, as well as the possibility to visualise changes over time. Regarding data provisioning, harmonisation and visualisation several methodological and technical approaches will be analysed and discussed.

For this research different datasets have been analysed, mainly from the Horowhenua area in the Manawatu-Wanganui (Horizons) region, New Zealand (Figure 1). Many of the provided files are in ESRI shapefile format (1). This format is widely supported and can store geographical features and associated attribute data in tabular form. Another important source of hydrological information in New Zealand is the database of the National Groundwater Monitoring Programme (NGMP) (2). Besides a wealth of groundwater quality properties, which are supposed to be representative for whole of New Zealand (Daughney and Reeves, 2006), groundwater level measurements are collected for the NGMP sites throughout New Zealand, too. The NGMP database schema is part of the GNS Geothermal and Groundwater Database (GGW (3)), which itself is a non-spatial Oracle© database. Additionally, 3D geological models (White and Reeves, 1999; White et al., 2011), i.e. layers, are provided in a xyz-tabular export from the

[1] ESRI shapefile format: http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf

[2] NGMP website: http://www.gns.cri.nz/Home/Our-Science/Environment-Climate/Water/Research/New-Zealand-Groundwater-Quality

[3] GGW database access: http://ggw.gns.cri.nz/ggwdata/

software EarthVision© as well as time-series from the region's state of the environment monitoring wells (required by the Ministry of Science and Innovation (MFE, 2012)) have been provided in comma-separated text form. The geographic coordinate system most widely used in New Zealand still is the New Zealand map grid projection (NZMG) based on the NZ geodetic datum of the year 1949 (NZGD1949). Newer datasets are now supposed to use New Zealand Transverse Mercator projection (NZTM2000) based on the NZ geodetic datum of the year 2000 (NZGD2000). Finally the WGS84 datum is used for web maps and GPS latitude longitude coordinates. Land Information New Zealand (LINZ, 2012) provides extensive information about the geospatial standards used in New Zealand. The datasets will be described in more detail later.

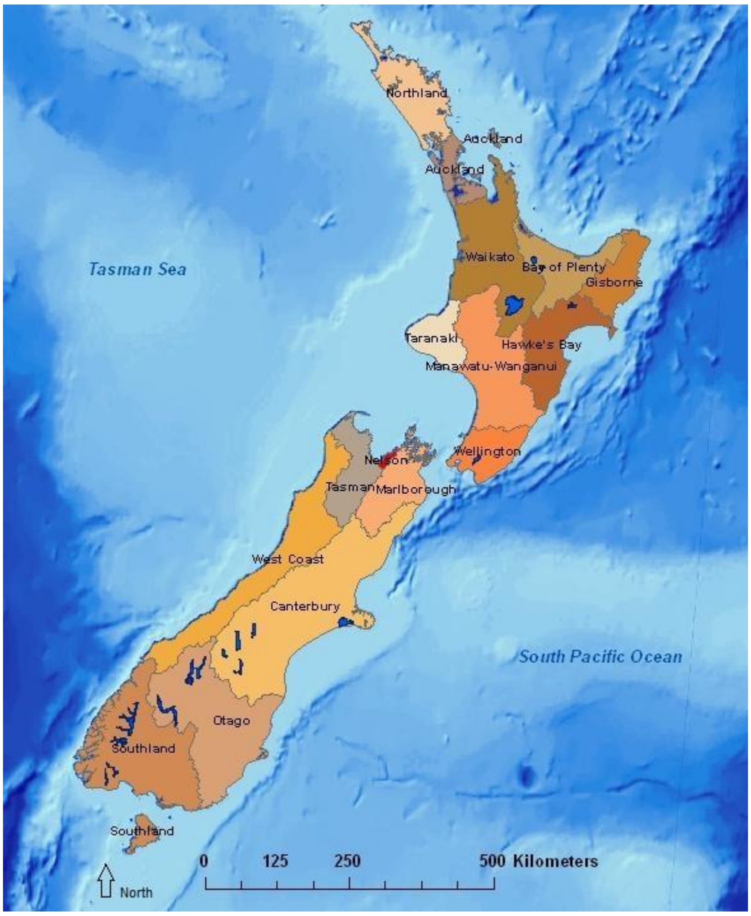

Figure 1: New Zealand and the regional councils (Kmoch et al., 2012)

To develop a web-based infrastructure that supports not only mapping of hydro(geo)logical datasets(Kessler et al., 2009), but also their effective visualisation in 4D space-time as well as the standards-based discovery of and access to the source datasets, existing (New Zealand and international) standards and best practices from reports of established successful projects are taken into account. Furthermore the integration into the developing New Zealand spatial data infrastructure (SDI, New

Zealand Geospatial Office [4]) is considered to be important. The NZ Geospatial Office declared national standards for describing and exchanging geospatial information that are based on OGC/ISO standards.

For a completely web-based architecture, the datasets should also be served via web services(Chang and Park, 2004). The Open Geospatial Consortium (OGC [5]) developed and published a set of standards how to structure, describe and deliver geospatial data (OGC, 2012b), many of them are also ISO standards. Most prominent established standards regarding data exchange formats is the XML-based Geographic Markup Language (GML) (OGC, 2007c) and the Simple Features specification (OGC, 2011c). To grant access to geospatial data, metadata and to visualise datasets, the following OGC web services (OWS) are well-known and will be implemented in the SMART portal:

- web mapping service (WMS) (OGC, 2006) and styled layers descriptor (SLD) (OGC, 2007d)
- web feature service (WFS) (OGC, 2010b)
- catalogue service for web (CSW) (OGC, 2007b)

Further important OGC web services and standards to share geospatial data include:

- web coverage service (WCS) (OGC, 2010a)
- web processing service (WPS) (OGC, 2007a)

To manage sensor networks and access sensor data and share time-series from observations in standardised encodings, following OGC standards exists and are already in use:

- sensor web enablement (SWE) (OGC, 2011b)
- sensor observation service (SOS) (OGC, 2012a)
- observations & measurements (O&M) (OGC, 2011a)

Several large scale projects incorporate OGC standards and have similar web-based approaches. In 2007 the INSIRE directive entered into force in Europe (Directive, 2007/2/EC). It describes a framework for a spatial data infrastructure across the European Union member states and heavily relies on OGC / ISO standards, too. The INSPIRE technical overview [6] describes the OGC WMS in version 1.3.0 as the INSPIRE view service. Furthermore a data download service and a discovery service is described in terms of a registry service and a service bus, which serve as the connecting and mediating

[4] NZ Geospatial Office, Geospatial Strategy http://www.geospatial.govt.nz/sdi-how-we-get-there/

[5] OGC website http://www.opengeospatial.org/

[6] INSPIRE technical architecture overview document from the JRC (Joint Research Centre) http://inspire.jrc.ec.europa.eu/reports/ImplementingRules/network/INSPIRETechnicalArchitectureOverview_v1.2.pdf

interfaces between the actual applications and geoportals and the data services. This concept is called service-oriented architecture (SOA) and is a software architecture pattern widely used in business applications. Main characteristics of SOA are the encapsulation of smaller applications, data providers and/or consumers as so called re-usable services and the orchestration of these services to larger applications to finally serve a use-case. The services are registered in a service registry and a service bus is connecting all the participating services again. Orchestration happens by exchanging data end sending messages over the service bus. In the last years research increasingly looks at the SOA pattern to effectively discover, access, process and visualise geospatial data in a completely online and web-based environment (Fils et al., 2009; Hildebrandt and Döllner, 2010; Huang et al., 2011). Besides the INSPIRE geoportal [(7)], also within the INSPIRE framework big projects like GENESIS [(8)] or GSSOIL [(9)] deployed geospatial information portals and incorporated OGC standards. The GENESIS project focuses on developing an innovative platform [(10)] not only to discover and access environmental datasets and databases, but also to integrate those with e.g. sensor networks and processing capabilities (Smolders et al., 2011). INSPIRE also describes a complete set of GML application schemas covering almost all aspects that are defined via the INSPIRE annexes. All participating European data providers are supposed to serve their spatial datasets conformant to the INSPIRE application schemas as well as providing sufficient metadata based on ISO 19115 (description of the spatial datasets or dataset series) and ISO 19119 (description of spatial services), whereas ISO 19139 describes the XML application schema to encode metadata content, e.g. within the OGC CSW web services context.

In North America two initiatives attract attention that on the one hand developed their own infrastructure and service to support their own needs, but subsequently could cooperate in the OGC groundwater interoperability experiment (Brodaric and Booth, 2011). One is CUAHSI [(11)], an American research organisation, whose members are mainly US universities. CUAHSI has done extensive works on massive exchange and processing of hydrological data and time-series across the United States. They developed a hydrological information systems, the CUAHSI-HIS [(12)] and the first version of WaterML, a XML-based application schema for the exchange of hydrological data. WaterML was not yet GML compliant. From the CUAHSI WaterML (v1.0 and v1.1) and the outcomes of the OGC groundwater interoperability experiment, the OGC hydrological

---

[7] INSPIRE geoportal: http://inspire-geoportal.ec.europa.eu/

[8] GENESIS FP7 project website: http://www.genesis-fp7.eu

[9] GSSOIL portal: http://gssoil-portal.eu

[10] GENESIS SOA architecture: http://www.genesis-fp7.eu/index.php/portalandservices-integration

[11] CUAHSI website: http://www.cuahsi.org

[12] CUAHSI Hydrological Information System (HIS): http://his.cuahsi.org/

domain working group (OGC HydroDWG [(13)]), which in turn also closely collaborates with the World Meteorological Organisation (WMO) and its Commission for Hydrology (CHy), developed WaterML2.0 that is near to official release as an OGC standard. WaterML2.0 in turn is based on OGC SOS 2.0 and OGC O&M 2.0 standards.

The other initiative is driven by Natural Resources Canada (NRCan [(14)]), a governmental organisation that is entrusted with the sustainable management of Canada's natural resources. The groundwater information network (GIN [(15)]) provides access to hydrological features and time-series data from eight out of the ten Canadian provinces. The architecture mainly comprises of the GIN mediator that harmonises the different input datasets and databases from the provinces and delivers harmonised datasets in the Groundwater Markup language encoding (GroundWaterML or GWML). GWML was developed by NRCan and is a GML application schema for the exchange of groundwater-related features as well as for observation. GWML(v1.1) is based on GeoSciML(v2.0) that already is an GML application schema for geoscientific data, but GWML is even more specific about the hydro(geo)logical domain, especially regarding the following information from the NRCan GWML website [(16)]:

- Aquifers and other kinds of HydrogeologicUnit
- Water Quantity, Flow system, Reservoir and Budget.
- Water Quality (natural quality), suspended, dissolved and colloidal content
- Water Wells, wells components, such as screens and casing

Besides the 2D spatial visualisation (web mapping) with OGC WMS and access to the datasets with OGC WFS and GWML output encoding, this Canadian groundwater portal also provides 3D visualisation of wells and boreholes. An enquiry for technical details has been answered and noted the use of VRML (ISO/IEC, 1997) for the 3D visualisation. VRML is a markup language to describe 3D scenes. It is the predecessor of X3D, are more modern 3D declarative markup language (ISO/IEC, 2008). Besides that, NRCan also took part in the development of WateRML2.0.

GeoSciML [(17)] is developed and governed by the international interoperability working group of the Commission for the Management and Application of Geological Sciences (CGI), which is a commission of the International Union of Geological Sciences (IUGS). GeoSciML has been the basis for the INSPIRE annex 2, geologic reporting and is used by

[13] OGC HydroDWG: http://www.opengeospatial.org/projects/groups/hydrologydwg

[14] Natural Resources Canada website: http://www.nrcan.gc.ca

[15] GIN portal: http://gw-info.net/

[16] NRCan GWML: http://ngwd-bdnes.cits.rncan.gc.ca/service/api_ngwds/en/gwml.html

[17] GeoSciML website: http://www.geosciml.org/

the OneGeology [18] portal that thrives to create a geological map of the whole planet, "… the target scale is 1:1 000 000 but the project will be pragmatic and accept a range of scales and the best available data."[19] Each participating country/organisation should export its geological map via OGC WFS and GeoSciML. Apparently GNS Science is serving the geological map of New Zealand with GeoSciML v2.0 as a check of the WFS response within the OneGeology portal reveals[20].

In Australia the remarkable AuScope [21] project was started. AuScope's mission is to provide an infrastructure, a platform, to incubate all knowledge regarding spatial data and technology concerning all of Australia. The Commonwealth Scientific and Industrial Research Organisation (CSIRO [22]) and GeoScience Australia [23], the two largest research organisations in Australia, as well as further Australian universities and governmental agencies are AuScope members. Within AuScope a Spatial Information Service Stack (SISS [24]) is developed to support Australian governmental and non-governmental organisations, agencies and research institutes to participate in an Australian spatial data infrastructure. Main building blocks of the SISS are following software components:

- Geoserver, a java-based open source data server that supports the publishing of spatial datasets with WMS, WFS
- Thredds, a java-based open source data server that supports the publishing of spatial datasets in WCS (e.g. supports netCDF format and OpenDAP protocol, too)
- FullMoon XML processing framework, to support the design and validation of further GeoSciML-based application schemas (based on their Hollow World model)
- Catalogue and vocabulary service, to develop controlled vocabulary and ontologies within GML/GeoSciML application schemas, based on SKOS/RDF languages and recommends the use of CSW

[18] OneGeology website: http://www.onegeology.org

[19] OneGeology – technical requirements: http://www.onegeology.org/technical_progress/technical.html

[20] "View GeoSciML" link in the OneGeology portal points to a GNS server: http://maps.gns.cri.nz/geology/wfs?request=GetFeature&typeName=gsml:MappedFeature&srsName=epsg:4326&featureID=1766,

[21] AuScope website: http://auscope.org.au

[22] CSIRO website: http://www.csiro.au/

[23] GeoScience Australia: http://www.ga.gov.au/

[24] SISS website: http://siss.auscope.org/

CSIRO also has done some exemplary work on sensor networks – the South Esk Hydrological Sensor Web in Tasmania, Australia (SOS Google Maps client ([25])). Furthermore CSIRO participated in the development of WaterML2.0 as member of the OGC HydroDWG. CSIRO and Geoscience Australia are also members of the CGI (IUGS [(26)]) and participate in the development of GeoSciML.

In the last years in New Zealand several larger, but distinct geospatial projects and initiatives emerged, based on the New Zealand Geospatial Strategy [(27)] that has been released by the NZ Geospatial Office (a department of Land Information New Zealand, LINZ).

The GeoNet project monitors earthquakes, volcanos and general seismic activity. Told from internal GNS staff, the whole instrumentation infrastructure from sensors to databases is actually not OGC SOS based. But as soon, as the data is available in the datacentre, it is made available via OGC SOS 1.0 with O&M 1.0 encoding for further processing and public visualisation in the web [(28)]. For that, GNS, who maintains the GeoNet project, uses the 52°North SOS server software [(29)] and the Postgresql database with the PostGIS extension.

Recently the National Institute of Water and Atmospheric Research (NIWA [(30)]) has launched an environmental information browser [(31)] for their metadata catalogue and sensor station catalogue for their vast amount of distributed climate and ocean monitoring sensors. In an interview with Dr. Jochen Schmidt, who is Chief Environmental Scientist at NIWA and responsible for the catalogue services, he confirmed that their catalogue services are built on the java-based OGC CSW 2.0.2 implementation Geonetwork Opensource [(32)]. Further investigation showed that NIWA implemented several metadata profiles/standards to be served by their catalogue. Besides special marine and meteorological profiles, their catalogue serves the New Zealand geospatial standard ANZLIC profile, which in turn is based on ISO 19115:2005 and ISO 19139:2007 metadata standards.

---

[25] http://www.csiro.au/sensorweb/au.csiro.OgcThinClient/OgcThinClient.html

[26] The Commission for the Management and Application of Geoscience Information of the IUGS http://www.cgi-iugs.org/

[27] NZGO Geospatial Strategy: http://www.geospatial.govt.nz/about-nzgo/

[28] GeoNet website: http://www.geonet.org.nz

[29] 52°North sensorweb community: http://52north.org/communities/sensorweb/

[30] NIWA website: http://www.niwa.co.nz

[31] NIWA Environmental Information Browser: http://ei.niwa.co.nz

[32] GeoNetwork Opensource catalogue service software: http://geonetwork-opensource.org/

Finally Landcare Research [33], another of the nine Crown Research Institutes (CRIs) in New Zealand develops a New Zealand-wide soil map with a nominal scale of 1:50,000. The project is funded by the New Zealand Ministry of Science and Innovation (MSI, 2012) called S-map and geoportal is accessible through the S-mapOnline website [34]. It is noted that the project's geoportal is built upon a free and open source software stack comprising of PostGIS, Mapserver and GeoTools and links to the Open Source Geospatial Foundation website (OSGeo [35]) for more information about open source geospatial software. The S-map project is also funded by the MSI (MSI, 2012) and same as for the SMART project, one of the S-map project requirements is to use free open source software and development processes.

The regional councils in New Zealand use quite a limited set of tools to manage hydrological data. From the results of the groundwater workshop at the 50th conference of the New Zealand Hydrological Society in December 2011 in Wellington, as well as from further workshops at regional councils, e.g. Environment Waikato, Horizons and Hawke's Bay regional council in the subsequent months, a list of the most widely used tools for management of hydrological data has been compiled (Table 1). Even if regional councils usually are working within in their regional boundaries and catchment delineation apparently is well aligned with regional borders, hydrological data exchange is necessary. This process usually comprises of interpersonal negotiation of the data exchange format and discussion about interpretation of parameters, procedures and values, before the dataset actually can be used.

Table 1: Tools used by regional councils with hydrological data (Kmoch et al., 2012)

Tool	Description
TiDeDa™	DOS/Windows-based database and reporting application for hydrology related time series data, uses special file formats, prequel to Hilltop, now maintained be NIWA http://www.niwa.co.nz/software/tideda-time-dependent-data
Hilltop Data Tamer™	Windows-based database and reporting application suite for hydrology related time series data, uses special file formats, server system which provides REST-style XML access, can import HydroTel™ data, OGC SOS 2.0 and WaterML2.0

[33] Landcare Research website: http://www.landcareresearch.co.nz

[34] S-mapOnline: http://smap.landcareresearch.co.nz

[35] OSGeo website: http://www.osgeo.org/

	support planned (successor of TiDeDa™) http://www.hilltop.co.nz/
Hydstra™	Database and reporting application for hydrologic time series data, Australian-based company was acquired by German-based company Kisters around 2003 that promotes Wiski and provides data migration path http://www.kisters.com.au/english/html/au/homepage.html
Kisters Wiski™	Full-fledged data management and reporting system for hydrological data and time series, Kisters worked on WaterML2.0 and took part in the OGC surface water interoperability experiment http://www.kisters.net/wiski.html Kisters OGC news http://kiwis.kisters.de/KiWIS/
HydroTel™	Telemetry / Sensor system from New Zealand based iQuest company (Hamilton), which was acquired by Kisters in 2007 http://www.iquest.co.nz/telemetry-systems-monitoring.php Kisters and HydroTel interoperability tests http://www.iquest.co.nz/environmental-monitoring-blog/index.php/2011/10/05/kisters-using-new-ogc-standards
Oracle™ and Spatial/Locator™	Multiple regional councils and agencies use Oracle database with and without its spatial extensions and implemented different, independent data models to store hydro(geo)logical data, bore, wells, springs, e.g. NGMP/GGW, EW bore database

It is to be noticed that the two mainly used software suites, i.e. Hilltop™ and Kisters™ are already supporting and/or planning support for OGC SOS and the WaterML2.0 format.

Regarding visualisation of spatial data in the web with the help of open source software, there are a couple of well-established web mapping tool and frameworks that also can be called software building blocks for a geoportal. The Open Source Geospatial Foundation (OSGeo [(36)]), one of the largest international hubs/incubators for open source geospatial

[36] OSGeo website: http://www.osgeo.org

software, presented a comparison of their main web mapping toolkits on the FOSS4G 2010 conference in Barcelona [(37)]:

- OpenLayers [(38)], a JavaScript library
- Mapbender [(39)], a PHP-based geoportal management software
- Geomajas [(40)], a Java-based, self-contained WebGIS
- MapFish [(41)], Python-based web mapping developer framework

In this listing OpenLayers has a special position, as it is a light-weight JavaScript framework, which itself is used as the viewer component in Mapbender and MapFish. OpenLayers completely works within the browser and has no server component. It can display WMS, WFS and WCS layers, apply styling and geographic transformations. Furthermore it has primitive support for OGC SOS and WPS.

The other three frameworks differ partially in functionality and operational use case, and all have server-side components where they communicate with dedicated protocols. Geomajas and MapFish are more developer-oriented and focus on the provision of an API to enable developers to create custom-made applications, tailored to the customers need. MapFish uses Python on the server-side, all browser functionality is based on JavaScript frameworks like OpenLayers and GeoExt. Geomajas incorporates the powerful Java GeoTools libraries in the server backend and uses an own set of sophisticated JavaScript tools to create a fluent user experience in its web client. The developers of Mapbender claim (refer to OSGeo web mapping typification, p.37 [(37)]) that their software package has “best of breed” support for OGC web services and is more end-user-oriented. It has a PHP server backend and relies on OpenLayers and JQuery as client component in the browser.

Still, all the introduced projects and frameworks mainly focus on 2D web mapping and there are more tools out there. Some use charts and graphs to display time-series. But in opposite to GIN, which uses VRML to visualise surface and well and boreholes, i.e. lithology and height profiles, none of the other projects’ established geoportals has a 3D visualisation component. The Monterey Bay Aquarium Research Institute (MBARI [(42)]) used VRML, respective GeoVRML, for massive bathymetry and oceanic data visualisation purposes. To play VRML scenes in a browser, a plugin is needed. Regarding

[37] OSGeo web mapping typification: http://www.slideshare.net/arnulfchristl/osgeo-web-mapping-software-comparison

[38] OpenLayers website: http://openlayers.org/

[39] Mapbender website: http://www.mapbender.org

[40] Geomajas website: http://www.geomajas.org/

[41] MapFish website: http://mapfish.org/

[42] MBARI website: http://www.mbari.org

MBARI's systems documentations they exploited the VRML97 and GeoVRML specification extensively and contributed to further developments. MBARI also considered a change to X3D in 2009 and proposed some extensions and changes in the X3D specification.

One of the SMART project's requirements is simple and easy access to datasets and the visualisation capabilities. An overview regarding (geospatial) 3D web technologies reveals a short list of possibly suitable candidates:

- VRML97, GeoVRML
- Flash- or Silverlight-based
- Java applet-based
- GoogleEarth
- X3D , X3D Earth (in combination with HTML5)

VRML97 (ISO/IEC, 1997) has a geospatial extension, to geographically reference locations within the 3D scene graph description. VRML is a declarative language with an own format. To view VRML files in the browser a dedicated plugin is necessary. Quite a variety of VRML plugins exist, all have their advantages and flaws regarding the visual representation capabilities and the exhaustive support of the VRML, respective GeoVRML specification.

The Adobe Flash© framework has a wide-spread market penetration. Numbers from different online surveys state that a version of the Adobe Flash player plugin is installed on 90%-99% of all PCs. But regarding the different versions of the installed Flash plugins there is great diversity, as Adobe regularly publishes updates to fix security problems and to increase functionality and performance. Flash is like a full-fledged application environment, supporting sophisticated controls and a large range of media types. Also there is strong support in 3D acceleration and the support 3D graphics cards performance features. Flash does not directly support any type of geospatial data types or functionality. Nevertheless Flash is often used to visualise 2D web maps with additional dynamic effects and a visually appealing user interface.

In the year 2007 Microsoft launched the Silverlight framework as a competitor to Flash in the field of rich internet applications. Since then Microsoft released several versions, with the latest supporting 3D acceleration, too. Silverlight can be technically compared to Flash in terms of supported functionality and graphically demanding user interfaces. Also it works as a plugin within the browser. There is no direct geospatial support, too. The market penetration is apparently less than Flash.

Another approach is the Java applet paradigm, which levels all functionality of the Java language in the web browser, which includes 3D support and full geospatial functionality based on included libraries that need to be loaded at start-up. Similar with Flash and Silverlight it works as a plugin in the browser. Additionally a Java runtime environment must be installed on the client computer. Again similar to Flash and Silverlight, the provided functionality is based on the version installed on the computer.

With GoogleEarth the Google company provides a virtual globe similar to their Google Maps product. Main functionality is based on visualising and geo-referencing places on top of the earth's crust, i.e. surface features like elevation, cities, and so on. It is possible to visualise and place own geospatial information within GoogleEarth via the Keyhole Markup Language (KML). An important point regarding the use of Google Maps and or GoogleEarth is that these are completely provided by Google and subject to change without notice. Furthermore license fees might be applicable depending on commercial or educational use and the intensity, i.e. number of request per particular time period.

In the recent years a successor of VRML has been developed – X3D, eXtensible 3D [(43)]. X3D is an XML-based markup language that describes 3D scene graphs in an open, human-readable and interchangeable format [(44)]. Same as VRML97 with GeoVRML, X3D has a geospatial extension called X3D Earth. X3D in version 3.2 is also an ISO-certified international standard (ISO/IEC, 2008). For most of its 3D objects like points or ElevationGrids, X3D Earth has a geospatial counterpart – GeoPoint, GeoElevationGrid and so on, which allows for geographical arrangement by spatial reference system and coordinates. Furthermore the upcoming, work-in-progress, internet standard of the HTML language in version 5 (HTML5) supports the inclusion of X3D natively. So to speak no plugin would be necessary to show 3D objects directly in the browser window. This is combined with other recent developments, namely the WebGL [(45, 46)] 3D acceleration interface, where the browser natively can access the 3D graphics cards acceleration features and the HTML5 canvas element, a versatile media display control in the browser. These features must be supported by the browser. Most modern browsers to date, except Microsoft Internet Explorer, already have at least experimental support for that. For Internet Explorer community plugins are available to upgrade that functionality.

Similar to former HTML(4) versions the HTML5 document structure is accessible via the document object model (DOM) and can be freely manipulated through JavaScript inside the browser. With x3dom (X3DOM, 2012), the German Fraunhofer IGD (Fraunhofer-

[43] X3D specifications, web 3D consortium website : http://www.web3d.org/x3d/specifications/

[44] Why X3D instead VRML: http://www.xml.com/pub/a/2003/08/06/x3d.html

[45] Wikipedia WebGL article and browser support: http://en.wikipedia.org/wiki/WebGL

[46] WebGL official website: http://www.khronos.org/webgl/

Institut für Graphische Datenverarbeitung IGD [(47)]) created an experimental open source, pure JavaScript framework with extensive X3D support that harnesses the power of X3D, HTML5 canvas and WebGL in an independent complete browser-based environment (Behr et al., 2009). Based on that combination several experimental and demonstrative approaches have evolved. Especially in the realm of city visualisations with the OGC CityGML standard a lot of work is done towards a standardised web 3D portrayal service specification (OGC, 2010c). These are mainly governmental and commercially driven city and building infrastructure web visualisation products (e.g. CityServer3D [(48)] and DeepCity3D [(49)]). Apparently the Fraunhofer Gesellschaft (i.e Fraunhofer IGD) and BRGM [(50)], who work on the latter project, are both Technical Committee Members at the OGC. There are some less public approaches towards a general 3D geospatial visualisation in the geospatial open source domains. The 3D community of the 52°North initiative [(51)] is in the process of open sourcing a geospatial 3D visualisation server that e.g. can read grid and DEM layers in various formats and would generate VRML output and/or render it and serve images analogous to a WMS service [(52)] following a rendering pipeline approach from the geospatial feature to a visual output (Reitz et al., 2009).

The main objective of this thesis is to integrate and harmonise existing hydro(geo)logical data sources to a common, open, and accessible data format. The hypothesis is that an application schema integrated into a designated web portal supports the access, discovery, processing and 3D / 4D (spatio-temporal) visualisation. To verify or falsify the hypothesis the workflow represented in Figure 2 will serve for the harmonisation and transformation of hydro(geo)logical data about New Zealand's ground- and freshwater systems. Special development objectives have been prioritised and with the exclamation marks in Figure 2. Furthermore a complete architecture for such a web portal will be designed.

---

[47] Fraunhofer IGD: http://www.igd.fraunhofer.de/Institut/Abteilungen/Visual-Computing-System-Technologies

[48] Fraunhofer IGD - CityServer3D: http://www.cityserver3d.de/

[49] DeepCity3D project website: http://www.deepcity3d.eu

[50] Bureau de Recherches Géologiques et Minières, France: http://www.brgm.fr/

[51] 52°North 3D community: http://52north.org/communities/3d-community/

[52] Triturus whitepaper and roadmap: http://52north.org/images/stories/52n/communities/3D/triturus%20white%20paper.pdf

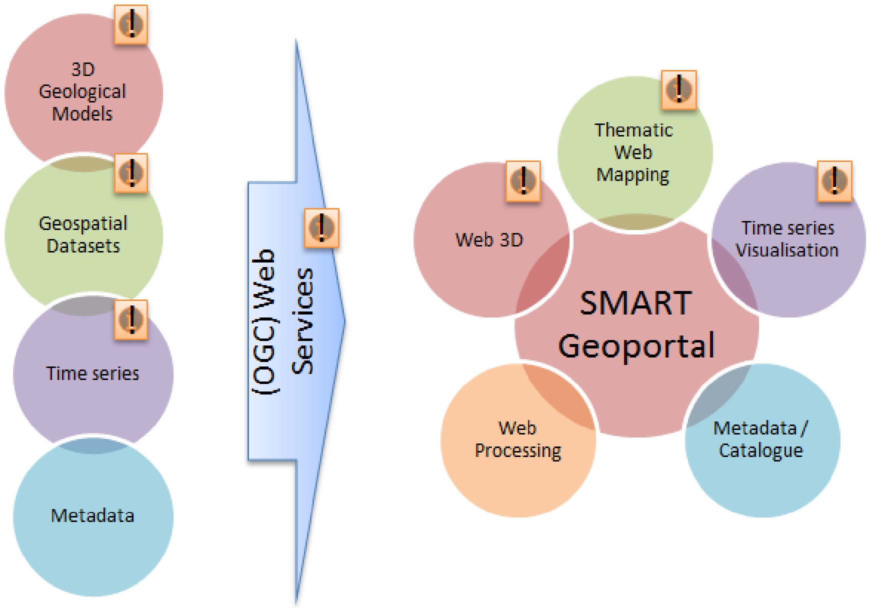

Figure 2: A conceptual model of the overall workflow, exclamation marks show implementations within this thesis

2 Methods

2.1 Data Harmonisation

As a first step to an integrated groundwater portal, all available datasets should be harmonised to allow a seamless view over whole New Zealand and enable meaningful comparison and analysis of datasets and geographical distribution patterns of distinct features and attributes. Therefore sample datasets have been collected and analysed (chapter 222.1.1). Regarding the nature of the collected datasets, from features over coverages to time-series, not all datasets need to be considered within the same classifications, e.g. different wells. Where possible the harmonisation approach tries to generalise and combine as many different datasets as possible.

Despite or even because of the plan of one distinct geoportal to access hydro(geo)logical datasets, the intention of an integration with the New Zealand SDI is still considered. So in the beginning a two-step technical harmonisation approach is developed. The first step would be for collected, scattered datasets that shall be directly loaded in to the provided geoportal data storage facilities (chapters 222.1.1 and 2.1.3). Secondly the harmonisation within the OGC interoperability concept is described, where datasets are made accessible through OGC web services in a coordinated document structure, also known as GML application schemas (chapter 2.1.2).

When the technical means are in place, a semantic harmonisation is necessary to have the same geographical references and vocabulary, i.e. terms, units and procedures used within the datasets (Atkinson et al., 2012).

2.1.1 Analysis of sample data – The Horowhenua water budget and geological model reports

The collected sample datasets are mainly based on a water budget report and a 3D geological model compiled by GNS for the Horizons regional council. All provided shapefiles, with no exception, use the New Zealand Map Grid (NZMG) projection. Important data classes are about hydrological features, like lakes, rivers, springs and wells. Also geology, soils and land use are within the contributed datasets. Just recently the complete quaternary geological map of New Zealand has been completed (QMAP 1:250 000). It can be assumed that as long as geology is represented in conformance to QMAP there won't be semantic harmonisation issues. It couldn't be evaluated, if the soil and land use classifications are valid around whole New Zealand. The naming of geographical features like lakes or rivers might be assumed to be consistent across New

Zealand. But there are place names, especially mountain summits that have an English and a Maori name.

Numbering and naming within the shapefiles is not consistent. Especially regarding the different datasets about wells and springs use e.g. “name”, “label”, “wellid” and “id” to reference wells. The elevation nomenclature is also not consistent, e.g. “Z” or “elevation” is used to name the corresponding column. For the groundwater level measurements mainly “GWL_(MASL)” is used, but also just “Z”. Therefore the context it needed to recognise, if the elevation of the wells or the actually measured value is meant. Figure 3 shows the established UML model for the analysed datasets.

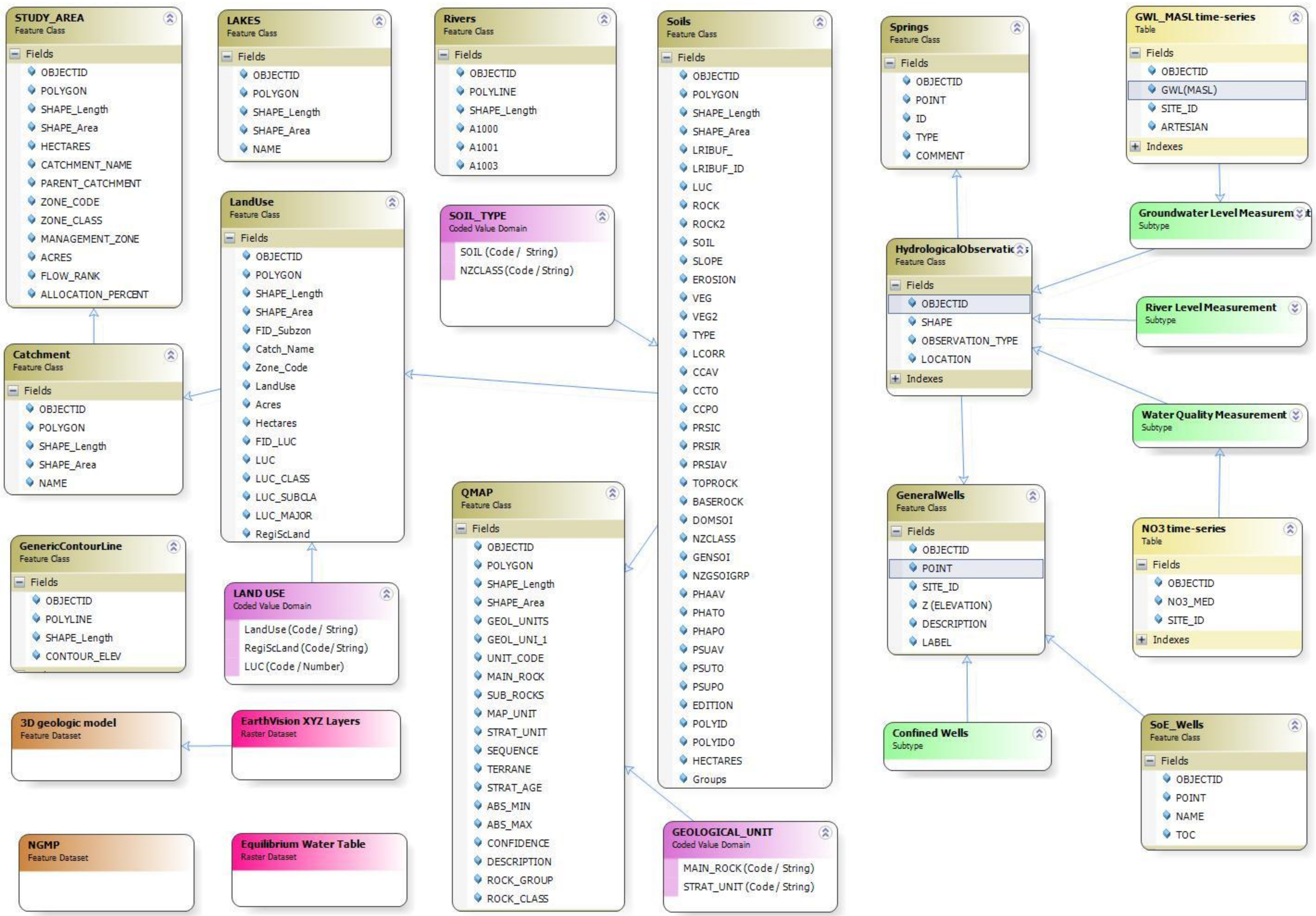

Figure 3: UML model representation of the provided shapefile, excel and raster datasets

In the left lower corner the NGMP feature dataset is mentioned. Within this thesis NGMP only has been used to extract groundwater level observations to be served as SOS/O&M. The NGMP database schema is based on the Australian National Groundwater Data Transfer Standard of 1999 (53). This data transfer standard (Figure 4) is quite generic as it does neither have a controlled vocabulary, nor a technical description of how the data has to be represented. Within GNS the model has been implemented as an Oracle© schema, but without spatial extension. That means geographic reference information is based on an implicitly assumed spatial reference system and coordinates are numbers in different columns. Additional changes have been made to support e.g. geo-thermal groundwater features, too. Within the NGMP the well construction elements are not generally used. Also the description of the well/groundwater feature surrounding (e.g. locality and site) is not consistently used.

---

[53] ANGDTS: http://www.brs.gov.au/land&water/groundwater/

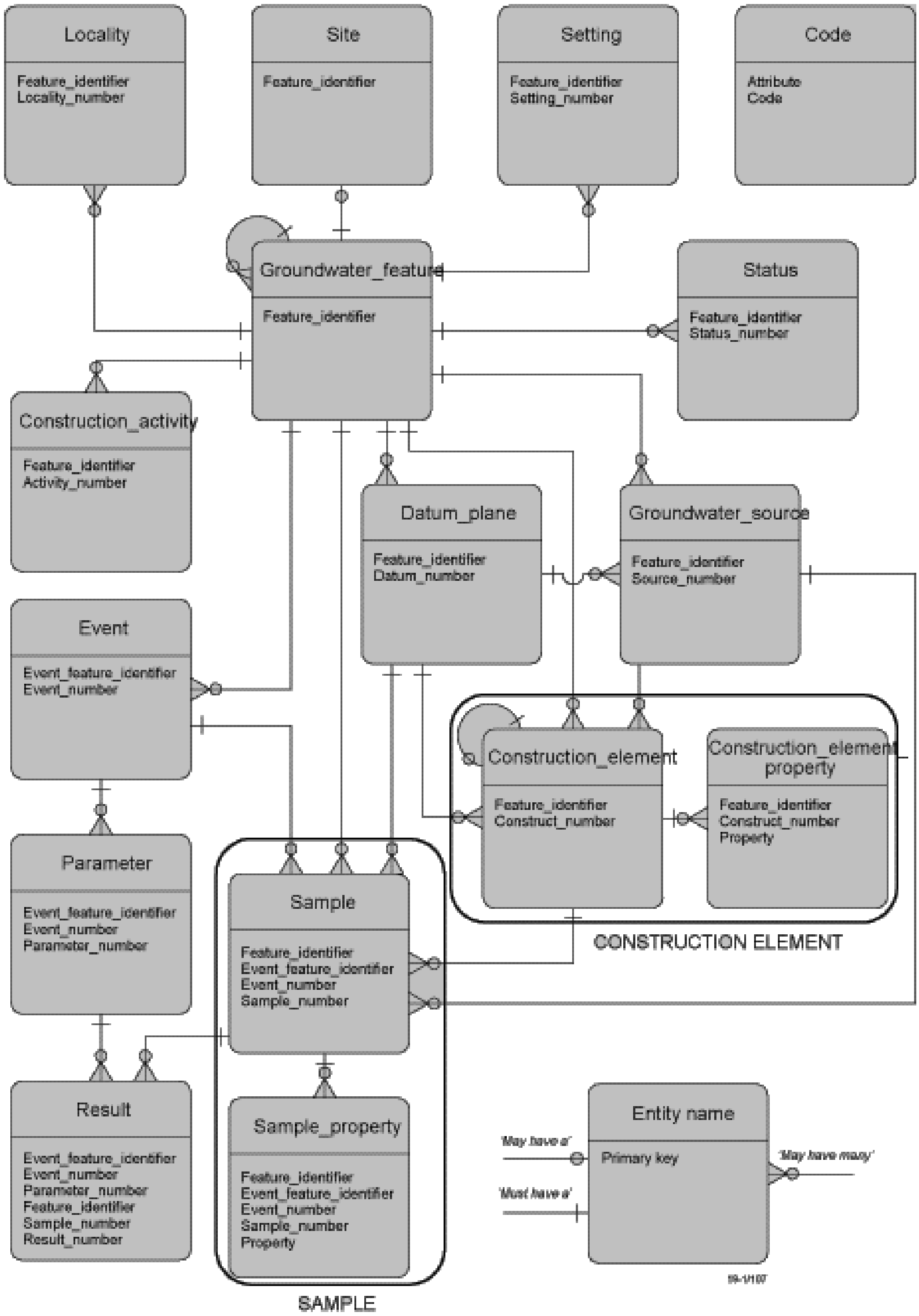

Figure 4: Australian National Groundwater Data Transfer Standard model (Australia, 1999)

The NGMP database schema is intensively used to collect and store (ground-) water quality, as well as water age, groundwater level measurements and further hydraulic properties. Well descriptions are detailed enough as it is necessary to find the well. NGMP even contains structures to describe aquifers and connect wells to aquifers and water bodies. But this feature is not used, probably because this information is not available in general. To deliver the NGMP based time-series, a 52°North SOS server has been customised (chapter 2.2).

The 3D geological model has been created within the software EarthVision© and the single geological layers have been exported as ASCII XYZ files. The points are described in NZMG easting/northing and a Z elevation value. Within EarthVision© the layers are based on a regularly spaced rectangular grid. With the export NODATA values within the grids are omitted, so that in the exported ASCII XYZ files only the data points from the cut-off area are available.

2.1.2 Abstraction and generalisation

The provided sample datasets are not sufficient to create a model that supports a generic aquifer characterisation. Within the regional councils apparently a more pragmatic, economic regime regarding groundwater, or generally speaking freshwater, seems to be in place. There are only visible and measurable features mapped. It would be a simple hydrological application schema only for wells, springs, rivers and lakes, and associated time-series observations. Still the main concern would be about the agreement on and the description of water quality and hydraulic properties and the procedures of their measurements. The NGMP database schema is probably the nearest in terms of provided fields for hydro(geo)logical and geographical features as well as necessary time-series support, but apparently it is only in use at GNS. Raster and DEM datasets as well as general imagery needs to be re-projected into one common projection. There is a discrepancy between the wide-spread use of NZMG and the more accurate newer NZTM, which is supposed to be used for new datasets. The decision might still be necessary to be evaluated with the stakeholders. Finally the question arises, how to include datasets that are not directly related to groundwater modelling, e.g. meteorological datasets or land use? Already available data services, e.g. provided by LINZ (LINZ, 2012) or the New Zealand meteorological service MetService MetConnect [(54,55)] could provide additional valuable data input for the SMART geoportal processing capabilities, Even if there is an obviously evident connection between meteorology, soil, land use and groundwater volume and quality, only directly groundwater-related datasets are considered in the following concept.

A first approach is to generate the database structure (SQL, 1st step technical harmonisation) and the GML application schema (2nd step technical harmonisation) from the established UML diagrams either by hand or more sophisticatedly with widely

---

[54] New Zealand MetService: http://www.metservice.com

[55] MetService MetConnect: http://www.metservice.com/services/metconnect

available modelling tools (discussed in chapter 4.1). This bears the risk of having a non-exhaustible schema, if not all existing and possible future datasets could be mapped into the generated schema. Extensions and further changes might be necessary. Most important for that approach would be intensive and wide-spread stakeholder consultation for a broad range of used datasets.

Based on the fact that the New Zealand geology is already available in a harmonised format (GeoSciML) a generic GeoSciML or GWML mapping approach for a hydro(geo)logical data portal might be adequate, instead of developing a completely new and independent application schema for hydro(geo)logical features.

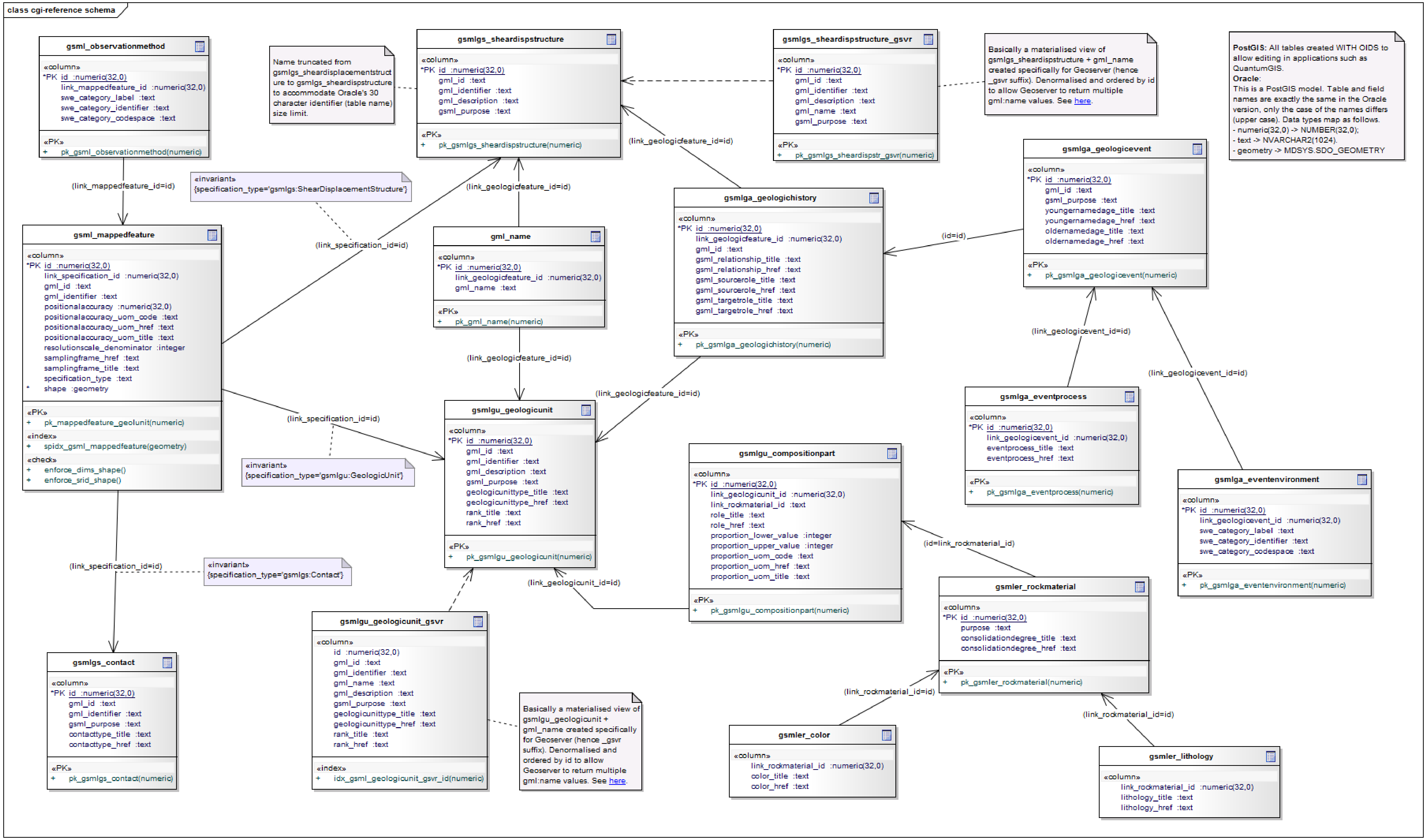

Figure 5: GeoSciML CGI reference model [56]

[56] GeoSciML CGI reference model from the SEEGrid website: https://www.seegrid.csiro.au/wiki/CGIModel/GeoSciMLReferenceDataset

GeoSciML is still quite generic and the central data type is the “Geologic Unit”. Also events, materials, mapped features and observations are existent as data type and do not need to be re-modelled explicitly, but can be mapped directly. A basic model approach would now need to map the features from the established UML diagrams into the GeoSciML feature classes and afterwards for the left-over features that could not be mapped into the existing GeoSciML schema, an additional GML application schema, which extends the GeoSciML schema would need to be created.

A well, spring or borehole are geological units. The time-series can be modelled as Mapped Features that have been observed by particular methods and that are related to events in time. Geological units can be comprised of several other geological units and have properties, like generic material and lithology. Aquifers and geological layers are obviously fitting in, too. The challenge with this approach is that it is still very generic and mainly designed for pure geological purposes, and an application, which should do actual processing based on the modelled feature classes would still need to be adjusted for different datasets. Therefore GWML has been created, which was exhaustively modelled for the groundwater domain. It is also documented, how the GWML schema complies with various ISO and OGC standards.

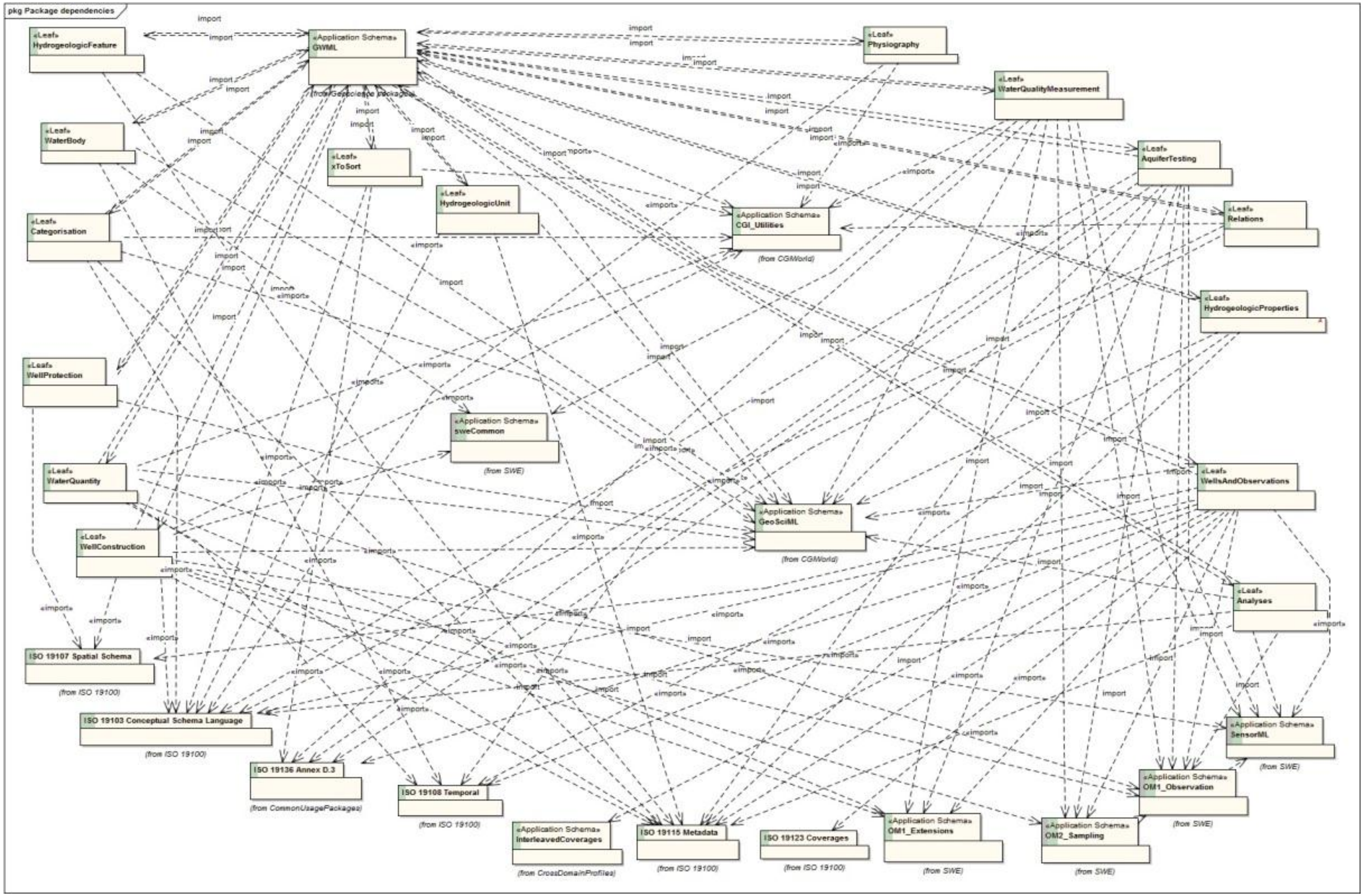

Figure 6: GWML 1.1 complete model (Boisvert and Brodaric, 2011)

Apparently all features from the Horowhenua UML model have respective representations in the GWML model overview (Figure 6). The schema files are also readily available on the NRCan website (Boisvert and Brodaric, 2011). A mapping would now be necessary from the underlying data source, which can be a database or other supported feature stores, to be served as an OGC WFS and additionally as SOS, if time-series of the observations are stored within the same data store/repository. However, the GWML schema in its recent version is outdated and a new version is developed that is aligned to the latest version of GeoSciML (v3.0) and GML v3.2.1.

The open source product Geoserver allows to map simple features from the underlying data store into complex models and application schemas [(57)] via mapping files, which describe, how the source data fields are combined into the complex features.

To harmonise hydrological observations and time-series, respective deliver harmonised hydrological time-series, the OGC HydroDWG has been working on the WaterML2.0 application schema that is designed for the use within SOS 2.0. WaterML2.0 is based on the generic O&M 2.0 schema and focuses on a more specific hydrological terminology. The inclusion of this topic into the SMART geoportal is described in more detail in chapter 2.2.

2.1.3 Manifestation in the database

Actually there is no dedicated necessity to have a standardised database model for a (OGC) harmonised data model, as every layer (single dataset) from the database can be mapped individually. Nevertheless would a proper database model strongly support the mapping process to the agreed-on harmonised application schema as well as the generally reasonable hydro(geo)logical data management within the SMART project. Also could the created database structure serve as a blue print for regional councils, who have a less developed IT infrastructure. Geoserver has strong support for a variety of data stores; amongst those is support for single shapefiles and Postgresql/PostGIS database layers. And it is possible to load shapefiles directly into the database with the PostGIS tool “shp2psql”. Either additionally necessary tables will be created in in the database to support all attributes from the shapefile or attribute fields from the shapefile can be manually mapped to existing database columns. So shapefiles with inconsistent attribute naming can be loaded into shared database tables. The pre-generation of data base tables based on a final application schema simplifies the mapping of the database feature into

---

[57] Geoserver app-schema documentation: http://docs.geoserver.org/stable/en/user/data/app-schema/index.html

the application schema features. Based on the provided Horowhenua datasets the database schema could be generated from the UML diagram. The different attribute names (e.g. column names in the database) of the well datasets could be streamlined, when loading the data into the tables. For more generic support within the GeoSciML application schema, a base SQL script is directly provided at the SEEGrid website (Figure 5). To prepare a database model directly based on the GWML application schema, an in-depth analysis would be necessary as it not always possible to transfer complex XML description into the more limited, constraint SQL language. But although the NGMP database schema is quite abstract, it is still modelled into fine details and shows similar structural elements like the GWML schema.

The OGC WFS-T specification, which is supported by Geoserver, allows for manipulation of features through the WFS interface. But the editing of features through WFS-T can only be done on layer basis, respective on simple features. There is not yet a possibility to directly edit complex features that a mapped together from simple features through the Geoserver app-schema mechanism described in the previous chapter 2.1.2. Within other geoportals data management is generally done from the backend side via dedicated update procedures and the only, relatively often changing data are actual time-series, which are not of a complex type and can be handled through the SOS-T (SOS transactional profile) if desired.

2.2 NGMP SOS

In comparison to WFS that allows direct data access on feature basis, SOS is part of the general OGC sensor web enablement (SWE) initiative and allows the aggregation of sensor data, sensor descriptions and observation time-series (Bröring et al., 2011). Sensor descriptions are encoded in SensorML and observation in O&M format, both are XML-based. To demonstrate the inclusion of standards-based time-series within the SMART portal and the general feasibility of OGC SWE within the hydrological domain, the NGMP dataset has been elected to serve as a use case. There are several OGC SOS compliant open source SOS servers available, amongst them the 52°North SOS server. The development community is very responsive and the documentation allows an easy start of an own customisation approach.

2.2.1 52North SOS server and the NGMP database

The 52°North SOS server is a Java-based, modular and robust software product, which is built with the help of the Maven2 build management framework. 52°North developers took part in the OGC groundwater and surface water interoperability experiments and the software is also used partly in the global runoff data centre (GRDC [58]).

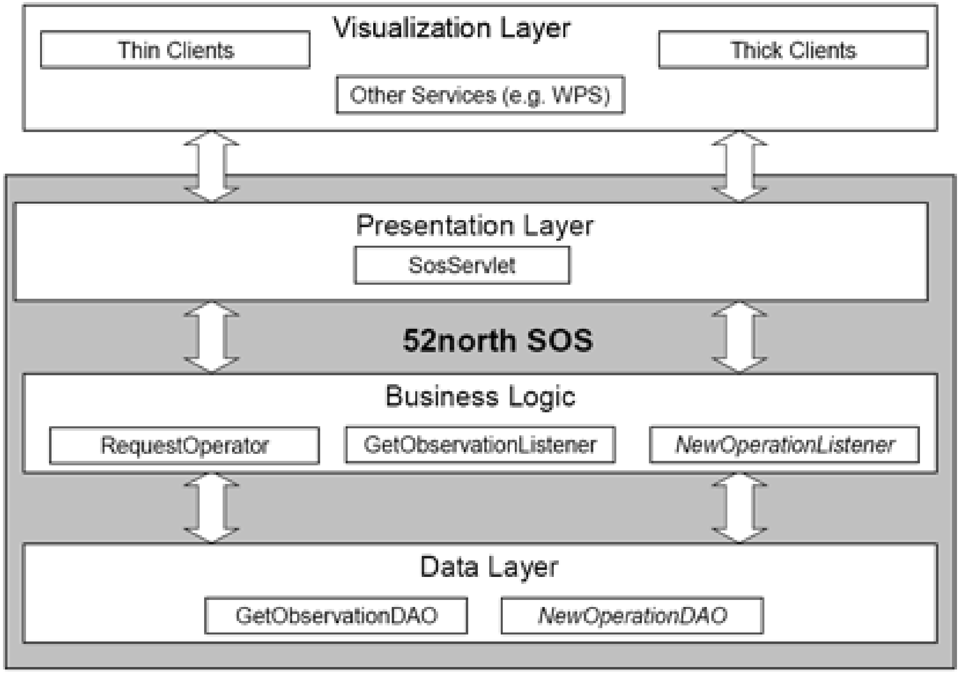

Figure 7: 52°North SOS software design [59]

Actually the modular architecture of this SOS server software (Figure 7) enables to implement a custom database connection by only implementing a different data access objects (DAO) layer. The OGC encoding classes and other web service routines don’t need to be touched. The SOS server has only support for the Postgresql/PostGIS database and comes with a dedicated database model Figure 8, which is optimised for the O&M encoding schema. For all supported SOS operations, e.g. the core operations are GetObservation, DescribeSensor and GetCapabilities, dedicated Java classes are to be implemented. As the NGMP Oracle© database does not have a spatial extension, the spatial queries are not supported. Based on the complex structure of the NGMP database model (Figure 4) a lot of joins across several database tables and views are necessary (e.g. Appendix A.2, Figure 15) to select the data that is necessary to serve GetObservation and GetFeatureOfInterest request.

[58] GRDC: http://www.bafg.de/cln_030/nn_299766/GRDC/EN/Home

[59] 52°North SOS software design: http://52north.org/communities/sensorweb/sos/design.html

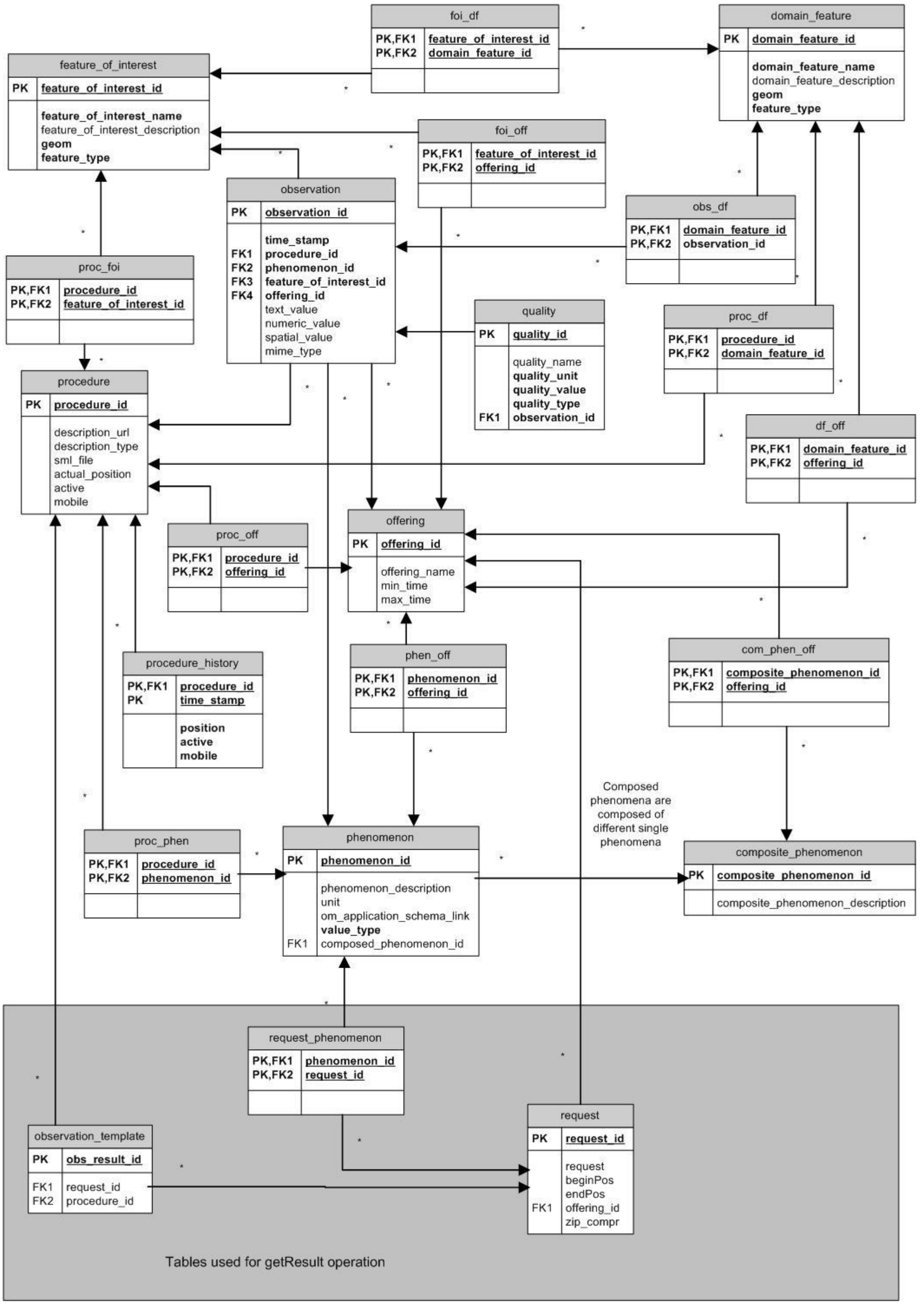

Figure 8: 52°North SOS server database model [60]

Mainly the SQL statements have been re-implemented to fit the NGMP database model to the SOS required data model (Appendix A.2, Figure 15).

There are many hydraulic and water quality parameters available in the NGMP. They are not all used for every sample that is taken from a NGMP monitoring well. The SOS

[60] 52°North SOS wiki ER-diagram: https://wiki.52north.org/bin/view/Sensornet/SosDataModeling

specification structures time-series delivery into general offerings, procedures and the observed phenomena. In the last years sensor networks are apparently used to access real sensor-based observations, so the offering is a generic name and could be "WATER_LEVEL" and the procedure usually would be a sensor, identified by a uniform resource name (e.g. "urn:ogc:object:sensor:system-xy123") and the observed phenomenon is identified by another reference (e.g. "urn:ogc:def:phenomenon:OGC:1.0:depth"). But the SOS and O&M specifications explicitly allow for "manual" observations, too. That means that the observed phenomenon or property could be collected/measured manually e.g. by taking water samples and send them to a laboratory for analysis. The procedure in this case would be modelled as a process instead of a sensor. But as this is not very common yet and for simplicity reasons regarding the general demonstration purpose, I modelled one legacy sensor, which is used for all sampling points, without describing further procedures.

Because of the complexity of the NGMP and the many interconnections and relations, I focussed only on the groundwater level values in the NGMP. Therefore I could hardcode the offering as "waterlevel" and the procedure as the mentioned legacy sensor and focus on the data retrieval. Two major incompatibilities arose regarding the re-implementation of the NGNP SOS DAO backend:

The time management needs to be adjusted, as there are either dates or times stored in the NGMP, but no complete timestamps, be it UNIX epoch seconds or an ISO completes datetime value, whereas the SOS only works with complete datetime timestamps based on the Postgresql datetime data type. The problem in that case is that the SOS specification describes the selection of observation data by time period. In the NGMP data structure each observation is linked to an event, and this event has a date (a day), a time and date/time reliability value, which describes how precise the date and time values of that particular event are. But this is a barrier to directly select the groundwater level result values by event time, because several conversions regarding time value formatting prevent the search result limitations via SQL filter mechanisms. That means all values are selected in the first run, and from that result set, the actually needed values are taken. This is not very efficient and needs to be addressed for a production deployment.

And secondly as the GGW/NGMP Oracle© database has no spatial extension, there is no native support for spatial queries in the form of "select all observation data from sampling sites within bounding box minX, minY, maxX, maxY". To still be able to deliver the geometry of the features (O&M: "sa:SamplingPoint"), where the observations have been sampled, the coordinates (GPS latitude and longitude columns in NGMP) and the implicit spatial reference system (WGS84) are used to construct OGC point features with the Java

Topology Suite (JTS) library. JTS provides a writer for the well-known-text (WKT) format for spatial data.

Finally the data collections from the DAO layer are transferred to the SOS internal encoder and transformed into the O&M XML format with the help of Java XMLBeans. The O&M XML schema files are generated into Java objects, which then can be filled with the SOS internal collections from the DAO layer.

2.2.2 A closer look on encodings

The general OGC XML format to serve observation time-series is Observations & Measurements 1.0 for SOS 1.0 and O&M 2.0 for SOS 2.0. The situation is similar to the GML feature application schemas. O&M is very generic. To support hydrological domain knowledge, WaterML2.0 has been developed from the insights and outcomes of the OGC Ground- and Surface water Interoperability Experiments. WaterML2.0 apparently has been passed general ratification in the OGC standardisation committee recently after a last workshop on ratings and gaugings by the OGC HydroDWG in Reading, UK, in June 2012. A WaterML2.0 encoding for the hydrological domain through a SOS server would be highly appreciable.

The 52°North initiative has been accepted as a tutoring organisation for the Google Summer of Code program and subsequently I have been accepted as a participating student to develop an exchangeable encoding system for the 52°North SOS server. After a simple CSV encoding plugin, I am actually working on a first WaterML2.0 encoding plugin, based on the most recent OGC WaterML2.0 schema. That would enable users of the 52°North SOS server to serve hydrological time-series in the WaterML2.0 schema. This is also remarkable, as the software producers of the main hydrological data management software suites for New Zealand also either have preliminary support, or advertise future support for WaterML2.0 (Table 1).

2.3 A 3D web visualisation concept for the Horowhenua area

As stated in the introduction, there are only very few possibilities to effectively implement 3D data visualisations in a web browser. Based on the idea of not having to rely on a particular plugin, an HTML5/WebGL approach seems reasonable. There are two possibilities, to exploit the WebGL functionality of modern browsers:

- X3D (with or without additional helper libraries)
- Native WebGL implementation

Generally speaking, WebGL exploits the low-level graphics routines of the operating system down to the computer's graphics card. Therefore a higher performance can be gained instead of in-browser rendering by using the sophisticated parallel shader and buffer routines of graphics processing units (GPU) of modern graphics cards. The WebGL interface is completely accessed via JavaScript libraries, which expose the low-level graphic routines to the browser and totally focus on providing an imperative ("how to do it") tool kit to generally compute all kinds of graphics.

X3D in turn is a declarative ("what you want") XML-based language, with which 3D scene graphs are a described. The approach is an empty scene, where events, objects and camera positions are declared. This makes it apparently easier to create a 3D scene or model, as the tool kit is already founded on the idea of a 3D world model. With X3D Earth geospatial components have been added, so that location can not only be described within an artificial Euclidean space, but also by a spatial reference system and coordinates. Furthermore does X3D come with functions and components to design an interactive user interface, also called HUD (heads up display), which simplifies the user's navigation within the created 3D space.

2.3.1 X3D surfaces

A particular X3D element has to be chosen to represent geological layers and surfaces (DEM etc.). X3D provides two main surface components for that task:

- ElevationGrid (resp. GeoElevationGrid for geospatial)
- IndexedFaceset

Main difference between these is that GeoElevationGrid has proper support for geo-referencing, but always assumes to display a fully rectangular, evenly spaced dataset. Only height values are used, because with the declaration of that element the equal point spacing has to be defined and the height values will be distributed line by line according to the defined numbers. Further configuration options describe so called viewpoints, the spatial reference system and texture, colours and lighting.

The IndexedFaceset is a more generic element that connects deliberately distributed points to polygon surfaces based on indexes. The points are indexed to define the surfaces they belong to and so a complex surface can be constructed though points belonging to two or more other surface polygons. Therefore points need to be defined for every polygon, which they belong to. Also there is no explicit support for true geospatial referencing.

2.3.2 Static vs. dynamic creation of a 3D web visualisation model

For demo purposes it might be valid to manually build such a 3D scene from the different source datasets – however in this case primarily the geological layers. It can be assumed that the number of available geological models will grow slowly and stay low, as it is quite an enormous effort to develop and build such a model, even for an experienced geologist or geophysicist. But the number of other features, like wells and bores should at some point also be included into the 3D web model. And not at last the demand to visualise actual near-real-time sensor data contradicts a static manual creation of such a 3D scene.

For future purposes it would be good to support a dynamic creation of preferably all kind of geospatial and time-series-based data (Kessler et al., 2009). Therefore a system needs to be designed that supports the aggregation of geospatial data and the transformation into a visual model. Based on the SDI paradigm and OGC web services and application schemas a generic rendering mechanism from GML- / O&M-based could be suggested. Another approach would be based on a particular 3D-spatially-enabled database (Breunig and Zlatanova, 2011). The disadvantage would be at least a doubling of data, as we still want to have the harmonised NZ hydrology or GWML and e.g. WaterML2.0 application schemas. But based on such domain-specific application schemas, a specific render pipeline could be designed (Reitz et al., 2009), as all available datasets comply with the agreed-on data structure.

So as the main feature data would come as GML, which is XML, an eXtensible Stylesheet Language Transformation (XSLT) can be developed to transform the GML-encoded features into an X3D scene. A general mapping would be necessary to transform GML point features to X3D GeoPoint elements within the target 3D scene graph, and so on. Imagery and raster data need to be handled separately – they would need to be blended in through additional functionalities.

There are initiatives going on regarding an OGC 3D web service (OGC, 2010c) and some demo implementations can be found in the web. The 52°North 3D community is one of the most promising candidates, not only in providing a static portrayal service complimentary to WMS, but also in online generation of real 3D model with VRML and subsequently X3D (Beard, 2006).

2.4 The WebGIS Architecture Model

Finally all building blocks need to be put into an architectural design to comprise a WebGIS or geoportal. The technologies and methods described in the former chapters are combined to a so-called stack. This stack is customisable and flexible and through the use of OGC web services it is also pluggable. The prototype WebGIS architecture (Figure 9) is now described in detail.

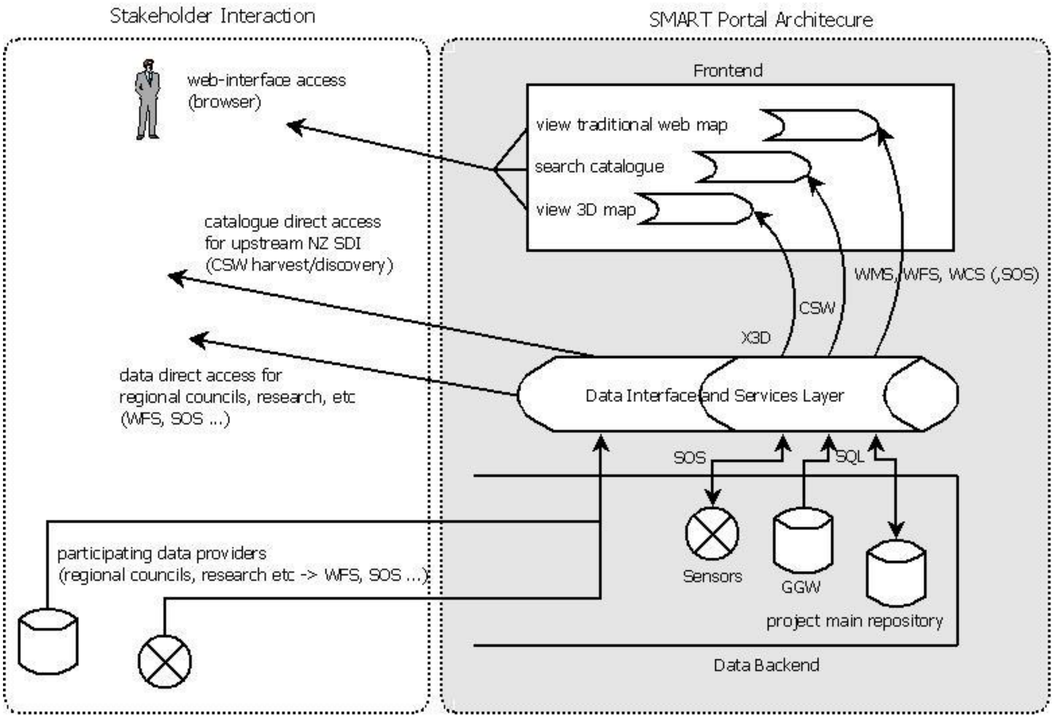

Figure 9: SMART portal architecture and stakeholder interaction (Kmoch et al., 2012)

2.4.1 Data repository

The foundation is a proper data backend. With the backend not only the actual database is meant, but also all means to store different forms of geospatial data, respectively where to access data that is to be visualised and/or processed within the geoportal. The main local part of the backend is the project repository, which includes a database, where most of the datasets are loaded into and a local files system, e.g. for satellite imagery and the server source base, i.e. the software installation directories. The database of choice is the latest Postgresql database v9.1 with the PostGIS extension 1.5 or respectively 2.0 for a test of experimental functionality, especially the raster support. Postgresql has strong support for database integrity through a sophisticated logging mechanism. These logging capabilities are also facilitated for remote database replication, which can be used to distribute read-only work load, as well as to enable higher reliability by creating standby database nodes. For security reasons the database can only be accessed on the localhost. The database TCP ports are bound to 127.0.0.1 and pg_hba.conf settings only allow a local user to authenticate. One main database schema is created based on the PostGIS template schema with the Postgresql database management system. The table structure is based on the harmonisation approach (chapter 2.1.3) but needs to be further refined for the Postgresql database system, e.g. to meet the correct data types and constraint declarations.

Furthermore all necessary and dynamic project-related data folders are moved into a consistent directory structure. Based on the Geoserver installation directories, a local data directory structure has been created and the Geoserver DATADIR parameter set to the particular top directory. This way Geoserver can build its working directories, the so-called workspaces, within the main data structure, which eases general file management, backup and recovery.

The NGMP database is regarded separately and is not counted to the main project repository. However its data can be accessed by the customised SOS server (chapter 2.2).

All other datasets are assumed to be served and consumed through OGC web services.

2.4.2 OGC web services

The basis of the used OGC web services is provided by a Geoserver installation. In comparison to the Mapserver [61], Geoserver supports an official application schema mapping in its WFS implementation. Datasets are served as WMS, WFS and WCS, whereas WFS and WCS mutually exclude each other based on the data source.

Time-series are suggested to be served and sourced only through SOS. An exception would be the inclusion of the Thredds server that may serve complex multi-dimensional netCDF datasets that in turn may include time-series, too. Through the Thredds WCS interface, “slices” or “arrays” from within the netCDF source files can be served as standard WCS coverage layer. The OpenLayers web mapping client supports SOS and WCS, whereas the SOS is usually shown in examples to place the sampling point features and WCS is used as a coverage/raster layer. But regarding the time-series and 3D visualisation, both services can be consumed and processed in the background and the graphical output is not limited to a web map representation. Instead graphs and more sophisticated visualisation methods are anticipated. Therefore SOS, especially with the

[61] Mapserver project website: http://www.mapserver.org/

domain specific WaterML2.0 encoding should be explicitly fed into large scale model-based processing and analysis methods within the pluggable web processing capacities.

52°North [62] or the ZOO project [63] implement a service framework that complies with the OGC WPS specification (OGC, 2007a). Regardless of their actual location in the network, they can be integrated seamlessly. Processing backends can be developed in native programming languages like Java or Python, but also the facilitation of the versatile R [64] or MathWorks MATLAB© [65] statistical and mathematical computing environments is possible.

To incorporate metadata based on OGC web services the integration of a CSW 2.0.2 (OGC, 2007b) server is suggested. Powerful implementations are Geonetwork Opensource [66], the ESRI Geoportal Server [67] and pycsw [68], which all comply with the CSW 2.0.2 specification and the ISO metadata and encoding standards 19115, 19119 and 19139. Geonetwork and ESRI Geoportal provide a full-fledged all-in-one catalogue solution including a fully functional and customisable web interface, where also end-users could be redirected to, to explore datasets and view and edit metadata. But based on a fully web services-based design paradigm, it also is possible to implement the less resource-hungry pycsw that only will be accessed through the CSW protocol. Metadata entry, search and general discovery services can be provided through the SMART geoportal frontend/user interface. The CSW response for a GetRecords or GetRecordById request is XML-based and complies with the ISO 19139 "application schema" for the catalogue service metadata response encoding and can be rendered e.g. into a valid human-readable HTML document using XSLT.

Based on the 3D visualisation approach the 3D geovisualisation service, e.g. a demo W3DS are also be considered an OGC web service that can be consumed within the WebGIS, i.e. displayed in the frontend this particular case.

2.4.3 Base architecture and frontend

The three-tier-design of the developed geoportal with data backend, services layer and frontend can be recapitulated in Figure 9. The services layer is yet considered more to be an abstract description of the overall aggregation of available web services that usually

---

62 52°North WPS: http://52north.org/communities/geoprocessing/

63 ZOO project WPS: http://zoo-project.org/

64 The R project for statistical computing: http://www.r-project.org/

65 MathWorks MATLAB: www.mathworks.com/products/matlab/

66 Geonetwork Opensource: http://geonetwork-opensource.org/

67 ESRI Geoportal Server: http://www.esri.com/software/arcgis/geoportal

68 Pycsw project website: http://pycsw.org/

serve data from an own storage or sensor network backend. These services do not necessarily be based within the SMART project's own infrastructure, quite the contrary is appreciated, to connect with regional councils, governmental agencies and further research institutes and NGOs to participate in a New Zealand SDI and to provide and consume services and integrate them into the SMART geoportal to support the efforts for a hydro(geo)logical one-stop-shop WebGIS. Then all registered services can be consumed and re-processed within the services layer, e.g. by WPS or (3D) visualisation services (Hildebrandt and Döllner, 2010). Finally all services could be plugged dynamically into the web frontend. The design of the services layer is to be regarded as quite important, as within such a service-oriented architecture also all the processing and analysis capabilities are transferred into the services layer. The advantage is the possibility of an integration of a distributed computing grid for high volume large scale numerical or graphical data processing (Giuliani et al., 2011; Pebesma et al., 2011).

The actual web frontend or user interface is a pure web application. Again based on the distributed web services paradigm, the web application can be based anywhere in the network/internet and just needs to be able to connect to the desired services. Based on the OpenLayers and ExtJS [69] libraries, a web mapping template, e.g. based on Z_GIS collaborative developments (Mittlböck et al., 2012) can be used to derive a demo web mapping application for the Horowhenua and NGMP datasets. Further dedicated developments would be necessary to create a registry for all supplied services and to adjust the web viewer application to dynamically load and unload required or unused web services. Also the catalogue capabilities need to be added to the web interface, too. It might be desirable to switch from the pure JavaScript-based web map viewer to an integrated web portal system (e.g. based on server-side Java or Python to stay in the programming language patterns of already used software components) and use the template only for the standard 2D web mapping purposes. The portal software can add flexibility and sophisticated configuration functionality regarding service orchestration as well as user and content management. It can also provide the viewing platform for advanced 3D/4D visualisations of spatio-temporal and geological datasets.

[69] ExtJS library on the Sencha website: http://www.sencha.com/products/extjs/

3 Results

Based on the developed overall methodology (Figure 2) and the portal design (Figure 9), for timely reasons only a subset of the developed methodology has been implemented within the thesis. The exclamation marks in Figure 2 reflect these parts:

The developed WebGIS and the NGMP SOS demonstrate general applicability of the designed portal architecture and underpin the harmonisation approach. A virtual machine instance, colloquially speaking the portal webserver, has been installed with the Ubuntu 11.10 Linux-based operating system in the Amazon Elastic Computing Cloud (EC2 [(70)]). Detailed important installation steps are listed in the appendix B.4. The basic operating system has been configured and software has been installed – the Postgresql database and the PostGIS extension, the Apache2 webserver, the ProFTPd ftp server and a Java runtime and development kit, which is necessary for most of the data servers, like Geoserver or 52°North SOS. An additional storage volume has been created and mounted into the portal server instance (mount point `/vol1`). All data folders for Geoserver and the Postgresql database have been moved and configured to use that new location in the file system. Also the Apache2 webserver root directory and ftp directory access are bound to that volume, too.

3.1 Harmonised and integrated existing data

The provided datasets (shapefiles and grid/raster files) have been analysed and an UML diagram has been created (Figure 3) with ArcGIS Diagrammer©. Unfortunately no useful export/generator functionality exists in that software, so the GML XSD application schema and the SQL could not be properly derived. Commercial and open source software tools exist that can generate a broad range of languages, but I have only seen commercial ones so far that provide integrated features to generate the XSD and SQL within one product. And with SQL usually ANSI standard SQL is generated, where no product specific extensions are considered. This is particularly challenging, as the spatial features like geometry representation are implemented slightly differently between database software vendors. Therefore the focus is on the general concept of such a particular workflow. The UML diagram needs to be re-implemented in a tool that supports the proper generation of XSD and SQL (preferably Postgresql/PostGIS in the case of the SMART portal). Chapters 5 and 6 will elaborate further on the outputs from the harmonisation approaches. The XSD schema files can then be used for the Geoserver

---

[70] Amazon EC2: http://aws.amazon.com/ec2/

app-schema mapping, which is exhaustively described in the Geoserver online documentation [71]. Remarkably, GeoSciML is used as example schema. The existent GeoSciML v3.0 and GWML 1.1 schema files are enclosed with the thesis, also the GeoSciML v3.0 based Postgresql/PostGIS SQL database schema.

Based on a 52°North SOS server v3.2 stable release, a custom DAO layer has been implemented. Mainly the SQL queries regarding data selection and the database driver have been modified. To support the different time handling and to create spatial features from the non-spatial NGMP database additional source code has been written. The full project source code is available on an additional DVD. Some implementation examples are shown in appendix B.1 (52°North NGMP SOS exemplary source code sections) and a reverse engineering database development screenshot in appendix A.2. At the time of the NGMP-SOS development WaterML2.0 was not yet released, therefore a WaterML2.0 time-series encoding demonstration is not part of this thesis anymore. However, with this student's participation in the Google Summer of Code project at 52°North an exchangeable encoding mechanism is now in development, which also tackles WaterML2.0 [72]. The latest reference application schemas in their XSD implementation for GWML v1.1, GeoSciML v3.0 and WaterML2.0 are added (Appendix A.3).

To support the geoportal web mapping demonstration all analysed 22 shapefiles have been loaded as data stores into the Geoserver. The NGMP well locations have been extracted from the NGMP database and loaded into the Postgresql database and are in turn published via WMS and WFS through the Geoserver's PostGIS data store support. For almost all layers customised Styled Layer Descriptors have been written, SLDs (OGC, 2007d), also referred to as "styles". Exemplary SLD code is listed in appendix B.3. The classifications for the groundwater level, evaporation and annual mean rainfall contours have been evaluated in ArcGIS© before.

3.2 WebGIS demo portal

The visible part can mainly be reduced to the map viewer or web mapping application. The map viewer template files have been deployed into the Apache2 webserver's document root directory. Also the necessary OpenLayers and ExtJS libraries have been downloaded and added to the Document root. The original MapViewer.js file has been modified. Two application contexts, "New Zealand overview" and "Horowhenua are" have been created and the Geoserver WMS layers distributed into these contexts

[71] Geoserver app-schema mapping: http://docs.geoserver.org/stable/en/user/data/app-schema/index.html

[72] https://wiki.52north.org/bin/view/Projects/GSoC2012ProjectsExchangeableEncodingsForSos

accordingly. NGMP sites, the Horowhenua study area outline were positioned into the first context and all other Horowhenua-related were put into the Horowhenua context. Based on these contexts, the layer tree and the legend window are only filled with the regarding layers, and therefore only those layers are visible in the map viewer window to allow a task-centred approach for the demonstration. GNS didn't allow or manage to enable a web access for the NGMP-SOS server, so two different demonstration approaches were necessary to enable a proper stakeholder presentation (Figures 10-12). Another customisation on the EnerGEO template was the introduction of a metadata/information button, which is position right of the layer in the layer tree window. The main idea is to enable direct access to the metadata belonging to that layer/dataset. In the demo case it is a generated URL that includes the layer name as parameter and displays a static webpage showing metadata.

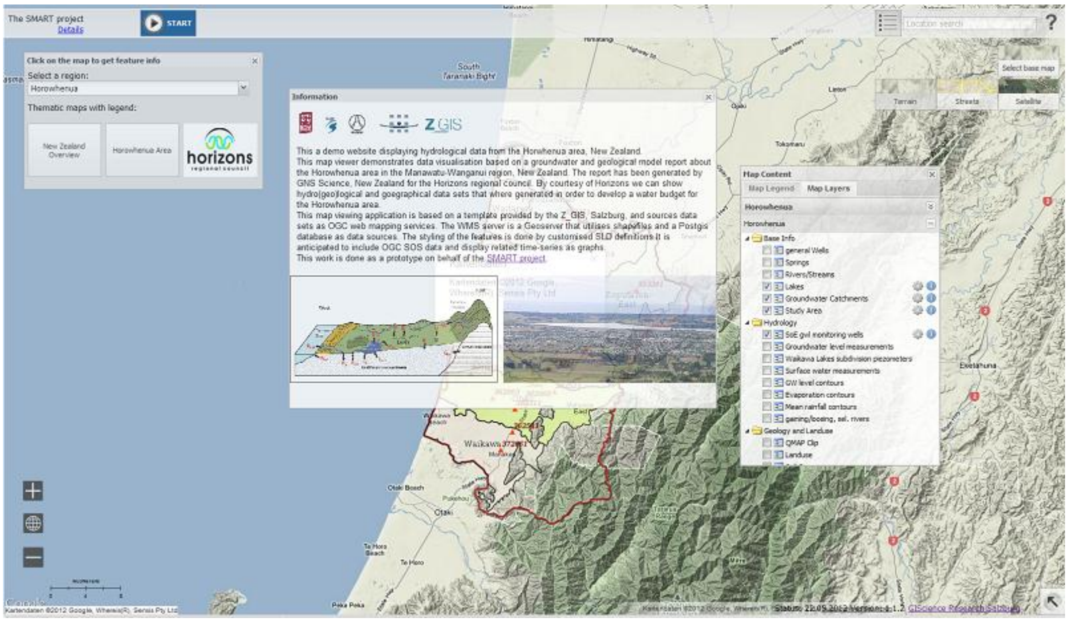

Figure 10: Web map viewing application, with introductory text and layer tree for the Horowhenua area

For the State-of-the-Environment (SoE) groundwater level monitoring wells an Excel sheet with time-series has been provided by the Horizons regional council. For these time-series, for each well within that dataset a hydrograph has been created and exported as a static picture with the well's id as the filename. The Geoserver's FeatureInfo template for the SoE wells layer has been customised (see appendix B.2) to load the particular hydrograph picture, based on the selected feature's well id property. So the time-series for the last 10-20 years for the selected SoE well could be visualised in a graph (Figure 11).

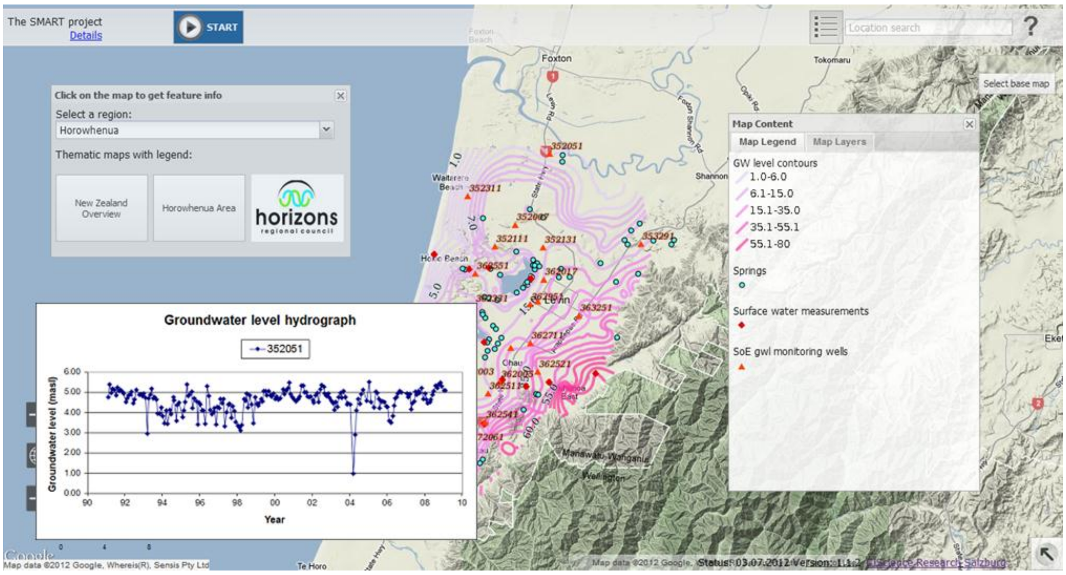

Figure 11: Groundwater level contours and the display of the hydrograph for a selected SoE well

As the NGMP-SOS was not yet published, a simple internal demonstration based on an OpenLayers SOS client and the Google Chart API [73,74] has been created. In the first request it queries the SOS GetCapabilities. In the response a list of all features of interest, i.e. the sampling points or NGMP sites are returned. Then the geometry is queried via GetFeatureOfInterest and the points a positioned on the map. With a click on one of the points a GetObservation request for that particular feature is issued and the response time-series data is drawn in the chart on the right side of the window (Figure 12).

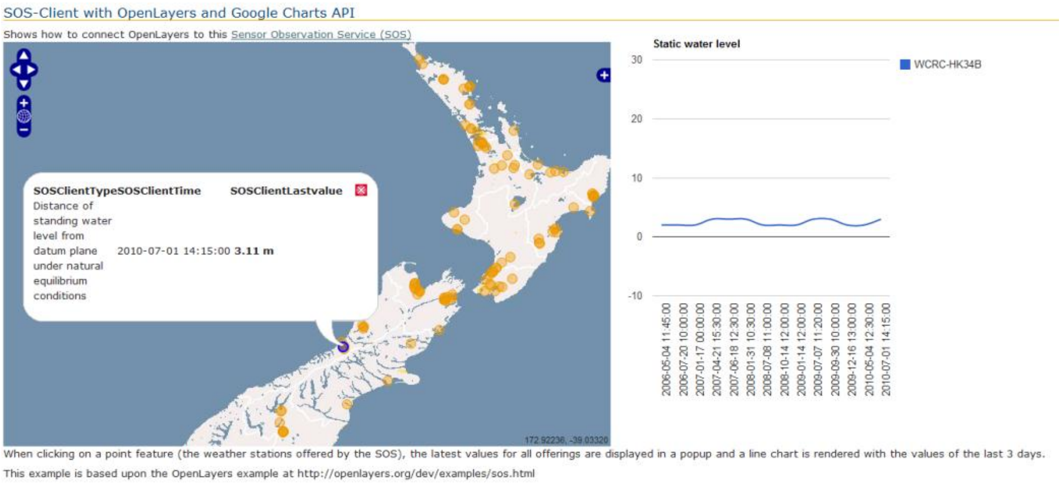

Figure 12: Simple OpenLayers and Google Charts SOS demo web client

---

[73] Example based on demo in the internet: http://ows.terrestris.de/examples/ol-sos-layer.html

[74] Google Charts: https://developers.google.com/chart/

3.3 X3D geological model

A small Java program has been written to transform the provided geological layers (ASCII xyz format, export from the software EarthVison©) into the elevation profile format of the X3D ElevationGrid specification. NODATA values were set to -9999. This distorts the overall relations, as the differences in height between the geological layers are within 300 meters. The x3dom JavaScript library has been used to integrate the developed X3D file into a simple HTML webpage (Figure 13). It could be viewed with the Chrome and Firefox browsers as expected. No further optimisations are functions have been applied.

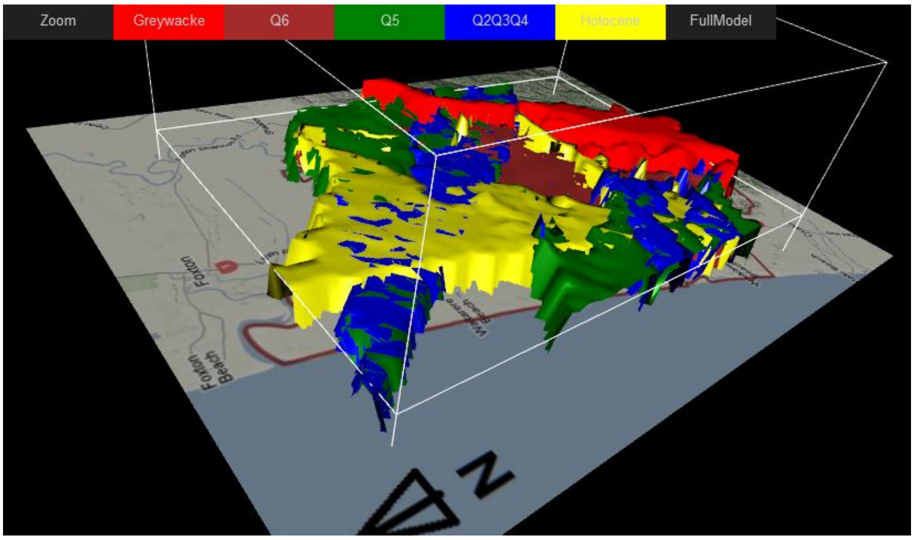

Figure 13: X3D ElevationGrids created from Horowhenua 3D geological model

4 Discussion

The demonstration implementation based on the introduced methodologies regarding harmonisation approaches (Figure 2) and architectural geoportal design (Figure 9) was rather straightforward. Based on the SMART project's requirement of preferring open source software, a workflow has described and partially implemented (Figure 14).

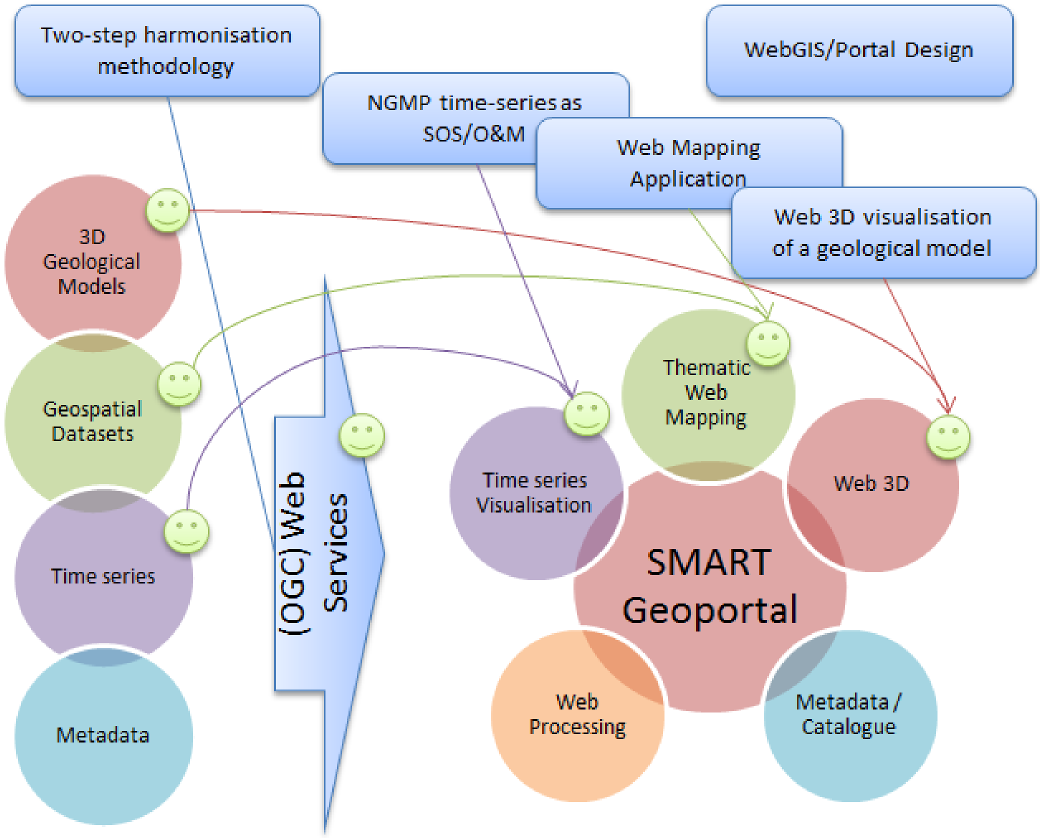

Figure 14: Based on the conceptual model (Figure 2) the outlined features have been implemented and demonstrate the general applicability of the overall methodology.

4.1 Data harmonisation

To use the ArcGIS Diagrammer© was probably not a good decision to develop an open UML model, as it didn't support any feature to export the UML model in a useful format. The automatic generation of schema files and SQL has the distinctive advantage of being idempotent and reproducible. Nevertheless manual refinement might be necessary, but an automatic approach is desirable. Apparently GeoSciML and GWML are developed using Sparx Systems Enterprise Architect© [(75)] as the UML modelling tool and schema generator. Based on that lesson more powerful tools like the Enterprise Architect,

[75] Sparx Systems Enterprise Architect©: http://www.sparxsystems.com.au/

Modelio, Umbrello, StarUML, XMLSpy or Eclipse Modelling Framework should be used to develop an overall New Zealand hydro(geo)logical information schema and pragmatically generate XSD schema files and the database schema for free use and publication for the stakeholders. That would eventually ease the interoperability efforts, if necessary schema information is freely and easily accessible for any interested party. Still there need to be some consideration about costs of the tools, their feasibility and eventually how to evaluate the use of a commercial tool regarding the project's requirements.

Regarding the choice, which harmonised schema to use, GeoSciML is not suitable, as it is too generic. Either an own New Zealand groundwater/aquifer/hydro(geo)logical information schema or e.g. GWML as a highly specific already available schema can be used. A recommendation is not easy. To develop an own schema can cost high efforts and much stakeholder interaction. As experience shows, that will need a lot of time resources as well. The other way round, to decide for a given, but foreign standard might also be a difficult decision to do for regional councils and other governmental stakeholders. The decision for a hydrological time-series interchange format on the other hand has been already made apparently. In recent stakeholder workshops and interviews a high interest in WaterML2.0 has been communicated. From this perspective the developments towards a 52°North SOS server with WaterML2.0 encoding support seem to meet stakeholder expectations (the latest reference application schemas for GWML v1.1, GeoSciML v3.0 and WaterML2.0 are added, Appendix A.3).

A general problem is the quality and diversity of the provided datasets. Most of the analysed datasets were based on two related reports and have been overworked and refined several times to suite the report needs. No lineage or metadata was actually available for these datasets. Therefore it was not really possible to seriously design a groundwater model. Nevertheless the general methodology is legit. It is important to communicate the importance of proper sample datasets not only within the project context, but also between stakeholders. This will be of even more importance, when a metadata standard is to be developed.

4.2 Technical portal implementation

The implemented WebGIS portal has demonstrated the feasibility of the developed concept. An independent web mapping application is sourcing and displaying WMS and SOS data. The versatility of Geoserver has been shown in providing support for different necessary data backends like Postgresql/PostGIS and shapefiles, as well as through creating custom styles (Appendix B.3) and feature information (Appendix B.2) for a proper and distinguished symbolisation of geospatial data in the web map. Also the comparison of Geoserver to other software tools emphasises the usability with a well sorted administration interface and application schema mapping support. Also the included WCS, WFS and WFS-T support can only be rivalled by suite of other open source tools, but not within one complete product. A disadvantage is the heavy resource usage of a Geoserver installation. The Amazon t1.micro virtual server instance with about 600MB RAM and one processing core was very slow and became unresponsive with too many parallel requests. For a production deployment a better performing server base is definitely recommended.

The software development for the 52°North SOS server took quite some time. The most challenging part was to get familiar with the software architecture, classes and methods and the general development pattern. But the Java programming language is well readable, powerful but still easy to learn. More important is the recapitulation of the object-oriented development style. More difficulties arose by exploring and learning the NGMP database schema. About 150 tables and another 100 views are existent. But not all of them are used anymore. The database schema apparently has already passed some evolutionary re-designs, where unused parts have not been removed – probably to be safe in case of legacy issues. The time handling needs to be reworked. I assume there is potential to support a correct datetime querying. Furthermore the spatial querying, needs a custom GeoTools facility needs to be implemented, which would build all known point location to feature objects, which then in turn could be queried in-memory, as the database does not support spatial queries. Finally the output encoding for WaterML2.0 could be added. That would mean to either backport the encoding plugin mechanism from the latest SOS server version into the v3.2 version, or better, to migrate the NGMP Oracle© DAO backend to the latest SOS server version.

The web mapping template after (Mittlböck et al., 2012) has no support for integration of a SOS layer. Furthermore all layers need to be hardcoded into the MapViewer.js file. These are two major barriers for a further use of the web mapping application as it is now. As the template is based on OpenLayers and ExtJS, the integration of a SOS layer might not be too difficult. With SOS 1.0 usually a WFS has been used to discover and serve features/sensors, whereas the SOS protocol has been used afterwards to actually retrieve time-series from that feature. With SOS 2.0 the feature handling can be completely done through the SOS protocol. And the integration of the web map viewer into a higher level web portal application and the decoupling of the manual layer configuration within the MapViewer.js file might be a possible solution to support a more dynamic application, where users can add and remove layers based on the catalogue searches.

The implemented metadata/information button in this development stage is only useful for demonstration purposes. The functionality can be extended to interlink the metadata, stored in a CSW catalogue, with the map viewer. Then a click on that button would actually request the latest metadata document for this dataset from a CSW catalogue service. The flexible web services architecture is designed for such interconnected operations. All mentioned services like WMS, WFS, WCS, CSW or SOS do not necessarily need to run on the same machine as the web portal application. That allows for work load distribution. But the major advantage is the interoperability with other data providers and/or consumers, whereas the legal constraints and the data ownership still lie on the provider side.

The 3D technology comparison is based on known circumstances and describes only roughly possible implementation concepts. An XSLT from GML to X3D approach could not be exhaustively examined. Also available 3D portrayal implementations could not be applied to the hydro(geo)logical domain. But the general use of the HTML5/WebGL technology through the use of X3D (and x3dom), could be demonstrated and is heavily recommended to implement an appealing interactive 3D/4D visualisation user interface.

5 Conclusion

The geoportal has been designed to support and visualise the scientific outcomes of the SMART project towards a New Zealand aquifer characterisation. Also interoperability concepts have been taken into account that enable the direct participation of the New Zealand regional councils as the main stakeholders. It has been shown how datasets can be harmonised, respectively how a harmonised information model can be developed. The harmonisation efforts are important, even necessary for automated application processing as well as for sharing styling/symbolisation definitions. This makes a seamless portrayal not only easier, but certainly possible in the first place, when different data sources about the same type of data come into play.

There are several possibilities to achieve a harmonised set of datasets. The preferred one is the one described in the thesis based on a direct publishing via OGC web services. Another way is to transform or convert datasets into the target schema. This would only be done in case of a data sharing request. This approach might be an intermediate step, if the establishing of a pure OGC web service approach cannot be implemented yet. And from this context the general data management paradigm needs to be reconsidered, within the SMART project and its portal on the one hand, and the regional councils own data management on the other hand. For the SMART project's research the additional hydrological data from the regional councils would be very valuable, so it would be good to incorporate their datasets, either through copying or through distributed online consumption and processing. The SMART project researchers could update own datasets either via WFS-T or through the backend. But if regional councils would need to make updates, they might either update locally and on the SMART portal or they would need to copy their local datasets again, convert it and then transmit a full new dataset, which is supposed to replace the old dataset. This means more efforts in data management and a higher risk of faults within the data processing.

The prototype portal architecture has been implemented, but for the project demands it needs to be enhanced as discussed. Possible workflow and harmonisation scenarios as well as supportive tools have been explained. A basic methodology has been derived inspired from best practises. But further investigation in stakeholder needs is necessary. It is to be accentuated again, how important the stakeholder interaction is for this kind of holistic approach to a New Zealand groundwater portal. And for that it is important to have central portal online even at an early development stage to demonstrate and to evolve it from the feedback with regional councils.

Within the SMART subproject “Data Synthesis and Visualisation”, where this thesis contributed to, the further research and developments in the next three years are to be continued by me as a PhD student. Therefore this thesis lays the headstone in the examination and dissemination about a New Zealand groundwater management portal and the means to analyse, characterise and visualise New Zealand’s aquifers.

6 Outlook

The data harmonisation procedure will influence the future data exchange standards in New Zealand. The careful crafting and implementation of such a national standard does not only require experience with the methodology and tools, but also familiarity with the regional councils, their data sets and usage as well as the cooperation with further data providers. Finally continuous stakeholder liaison is crucial to establish standards. With WaterML2.0 a strong candidate, soon to be released as an OGC standard for time-series encoding, is brought into play. With the early support for this standard within the SMART portal and the collaboration, not only with possible data owners like the regional councils, but also with locally established software providers like Hilltop™ we can take part on the forefront of establishing useful data exchange standards, which in turn opens a wealth of hydrological datasets for the SMART project's research towards aquifer characterisation.

The NGMP database, respectively an updated SOS (supporting WaterML2.0) has the potential to be the first data service to deliver via a new standard and to serve as a reference. And if the web mapping and/or processing functions in the WebGIS would support WaterML2.0 as well, the SMART portal would be the first reference application that actually consumes/visualises hydrological time-series in that new international standard.

It is desirable in the future to access detailed feature data from the NGMP. A Geoserver deployed with the general harmonised application schema can be used, to map the tables into the schema and serve them as layers in WMS and WFS. But as the NGMP Oracle© database has no spatial extension, a modified Geoserver data store must be implemented. The developments could be open-sourced and contribute to the geospatial community. Another approach would be the use of the ESRI ArcSDE© tool, which is established at GNS. It can expose NGMP Oracle© tables as layers, which in turn can be read by Geoserver.

Regarding the decision for a harmonisation scheme the possibility should be evaluated, to take part in the international development of a generic groundwater data application schema. NRCan is promoting the further development of GWML to an OGC standard. Therefore the best practises and experiences from GWML v1.1, GeoSciML v3.0 and the INSPIRE application schemes are to be combined and refined to design a new advanced application schema as an international standard.

Finally GIScience catch words of the last years as “Big Data”, “SOA” or “geovisualisation” and fancy trends like NoSQL, geospatial data mining and interactive 3D can be considered when thinking about the SMART data synthesis and visualisation efforts. And besides a functional geoportal with the outlined features, the whole project has the potential to not only bring up a technically sophisticated piece of highly integrated software that meets all stakeholder needs and establishes a new era of hydrological data management in New Zealand, but also to become a visually appealing master piece of “geovisualisation”.

Appendix A: Source data

A.1 Horowhenua report raw data field lists

(Excel file added to DVD)

A.2 NGMP database used schema tables and views

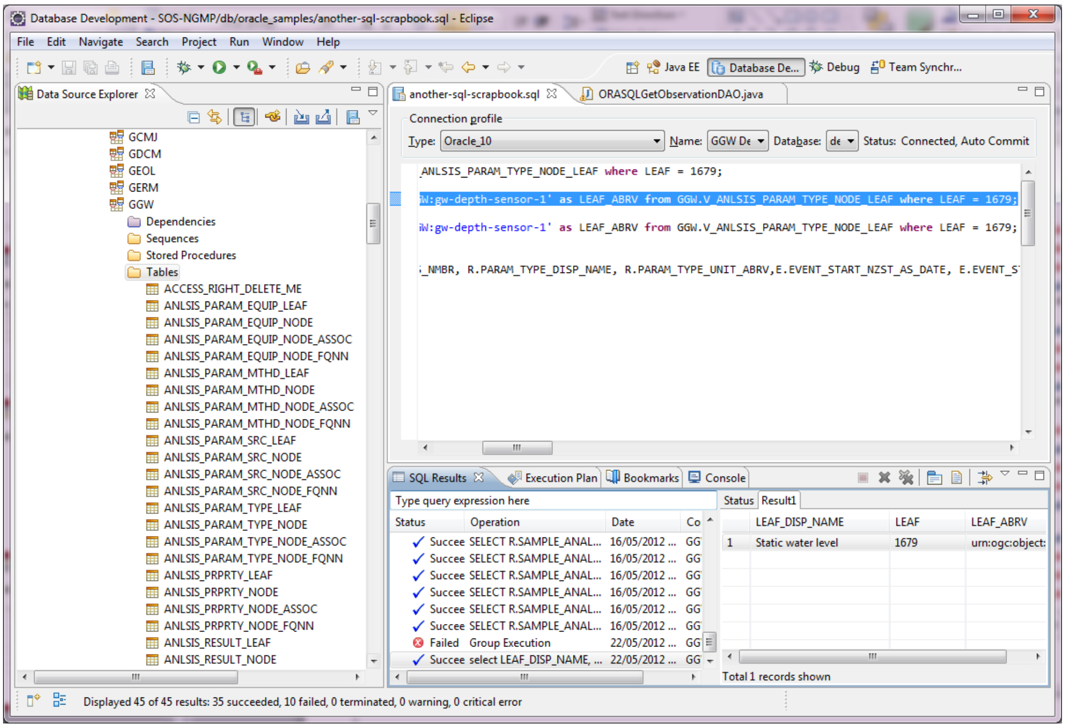

Figure 15: Screenshot of SOS database connection development

(full list from NGMP-SOS code provided on DVD)

A.3 GeoSciML, GWML and WaterML2.0 reference schemas

(full schemas added to DVD)

Appendix B: Code Snippets

B.1 52°North NGMP SOS exemplary source code sections

1. Select latest observation for given sampling point(s), GetObservationDAO.java

```
SELECT ROWNUM, SAMPLE_ANALYSIS_RESULT_PUB_ID, RESULT, RESULT_AS_NMBR,
PARAM_TYPE, PARAM_TYPE_DISP_NAME, PARAM_TYPE_UNIT_ABRV, FEATURE,
EVENT_START_NZST_AS_DATE, EVENT_START_TSTZ_AS_DATE FROM
            (SELECT R.SAMPLE_ANALYSIS_RESULT_PUB_ID, R.RESULT,
R.RESULT_AS_NMBR,
            R.PARAM_TYPE,    R.PARAM_TYPE_DISP_NAME, R.PARAM_TYPE_UNIT_ABRV,
E.FEATURE,
            E.EVENT_START_NZST_AS_DATE, E.EVENT_START_TSTZ_AS_DATE
                      FROM GGW.V_EXP_SAMPLE_ANALYSIS_RESULT R
                      RIGHT OUTER JOIN GGW.V_EXP_SAMPLE_ANALYSIS A
                      on (R.SAMPLE_ANALYSIS = A.SAMPLE_ANALYSIS)
                      RIGHT OUTER JOIN GGW.V_EXP_SAMPLE S
                      on (A.SAMPLE = S.SAMPLE)
                      RIGHT OUTER JOIN GGW.V_EXP_EVENT E
                      on (S.EVENT = E.EVENT)
                      WHERE (R.PARAM_TYPE = " + offering + ") AND (E.PROJECT =
10022) " + extraFoiFilter.toString() +
                      ORDER BY E.EVENT_START_NZST_AS_DATE DESC )
             WHERE ROWNUM = 1;
```

2. Select features, GetFeatureOfInterestDAO.java

```
SELECT FT.FEATURE, FT.FEAT_DISP_NAME, FT.FEAT_TOTAL_LENGTH, FT.FEAT_DESCR,
                  FT.FEAT_LOCATION_DESCR, FT.FEAT_CALC_NZGD1949_LATITUDE,
FT.FEAT_CALC_NZGD1949_LONGITUDE,
                  FT.FEAT_CALC_NZGD1949_LATLONG_SRC,
FT.FEAT_TYPE_LEAF,FT.FEAT_IS_VALID, FT.FEAT_IS_CNFDTL
                   from GGW.FEATURE FT
                   RIGHT OUTER JOIN GGW.PROJ_FEAT PF
                   ON (FT.FEATURE = PF.FEATURE)
                   WHERE (FT.FEAT_IS_VALID = 1) and (FT.FEAT_IS_CNFDTL = 0)
AND (PF.PROJECT = 10022) AND (FT.FEATURE = ' + featureOfInterestIds[i] + " ");
```

3. Maven development structure

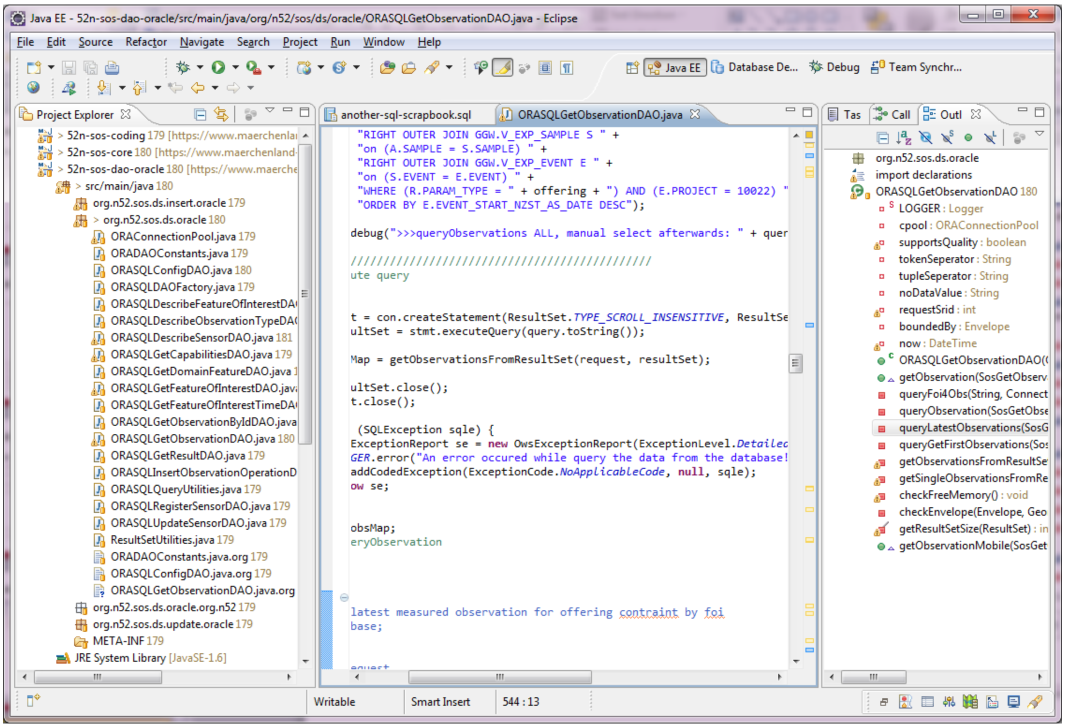

Figure 16: Screenshot of DAO backend development and Maven module structure

(full source provided on DVD)

B.2 Geoserver custom FeatureInfo

SoE wells hydrograph

```
<#--
Body section of the GetFeatureInfo template, it's provided with one feature
collection, and
will be called multiple times if there are various feature collections
-->
<table class="featureInfo">
  <caption class="featureInfo">SOE Wells</caption>
  <tr>
<#list type.attributes as attribute>
  <#if !attribute. isGeometry>
    <th >${attribute.name}</th>
  </#if>
</#list>
  </tr>

<#assign odd = false>
<#list features as feature>
  <#if odd>
    <tr class="odd">
  <#else>
    <tr>
```

```
  </#if>
  <#assign odd = !odd>

  <#list feature. attributes as attribute>
    <#if !attribute.isGeometry>
      <td>${attribute. value}</td>
    </#if>
  </#list>
  </tr>
  <tr>
        <td colspan="4">
                <img src="demorequests/${feature. attributes["ID"].value}.png"
width="400" height="216" alt="SOE Well ID ${feature. attributes["ID"].value}
Hydrograph">
        </td>
  </tr>
</#list>
</table>
<br/>
```

In the JavaScript implementation of your OpenLayers map-object you fetch this template "auomagically" filled with data from Geoserver. You need to patch together following things:

```
  featureInfo = OpenLayers.Control.WMSGetFeatureInfo()
        map.addControl(featureInfo)
        featureInfo.activate()
        document.getElementById('map').style.cursor='pointer';
```

And finally you could display that generated html via an Ext.Window

B.3 Geoserver custom SLDs

1. SoE triangle with label

```
<?xml version="1.0" encoding="UTF-8"?>
<StyledLayerDescriptor version="1.0.0"
  xsi:schemaLocation="http://www.opengis.net/sld
http://schemas.opengis.net/sld/1.0.0/StyledLayerDescriptor.xsd"
xmlns="http://www.opengis.net/sld"
  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>soe_gwl_monitoring_wells</Name>
    <UserStyle>
      <Name>soe_gwl_monitoring_wells:withLabel</Name>
```

```
    <Title>withLabel</Title>
    <FeatureTypeStyle>
      <Name>triangleWithLabel</Name>
      <Rule>
        <Name>triangleWithLabel</Name>
        <MinScaleDenominator>0</MinScaleDenominator>
        <MaxScaleDenominator>9999999</MaxScaleDenominator>
        <TextSymbolizer>
          <Label>
            <ogc:PropertyName>ID</ogc:PropertyName>
          </Label>
          <Font>
            <CssParameter name="font-family">Sans-Serif</CssParameter>
            <CssParameter name="font-style">italic</CssParameter>
            <CssParameter name="font-size">10</CssParameter>
            <CssParameter name="font-color">#000000</CssParameter>
          </Font>
          <LabelPlacement>
            <PointPlacement>
              <AnchorPoint>
                <AnchorPointX>
                  <ogc:Literal>0.0</ogc:Literal>
                </AnchorPointX>
                <AnchorPointY>
                  <ogc:Literal>0.0</ogc:Literal>
                </AnchorPointY>
              </AnchorPoint>
              <Displacement>
                <DisplacementX>
                  <ogc:Literal>2.0</ogc:Literal>
                </DisplacementX>
                <DisplacementY>
                  <ogc:Literal>2.0</ogc:Literal>
                </DisplacementY>
              </Displacement>
              <Rotation>
                <ogc:Literal>0.0</ogc:Literal>
              </Rotation>
            </PointPlacement>
          </LabelPlacement>
          <Halo>
            <Fill>
              <CssParameter name="fill">#FF4000</CssParameter>
              <CssParameter name="fill-opacity">0.3</CssParameter>
            </Fill>
          </Halo>
          <Fill>
            <CssParameter name="fill">#000000</CssParameter>
          </Fill>
        </TextSymbolizer>
        <PointSymbolizer>
          <Graphic>
```

```
              <Mark>
                <WellKnownName>triangle</WellKnownName>
                <Fill>
                  <CssParameter name="fill">
                    <ogc:Literal>#FF4000</ogc:Literal>
                  </CssParameter>
                </Fill>
              </Mark>
              <Opacity>
                <ogc:Literal>1.0</ogc:Literal>
              </Opacity>
              <Size>
                <ogc:Literal>10</ogc:Literal>
              </Size>
            </Graphic>
          </PointSymbolizer>
        </Rule>
      </FeatureTypeStyle>
    </UserStyle>
  </NamedLayer>
</StyledLayerDescriptor>
```

2. GWL contour line

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<StyledLayerDescriptor                                          version="1.0.0"
xsi:schemaLocation="http://www.opengis.net/sld       StyledLayerDescriptor.xsd"
xmlns="http://www.opengis.net/sld       xmlns:ogc="http://www.opengis.net/ogc"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <!-- a Named Layer is the basic building block of an SLD document -->
  <NamedLayer>
    <Name>groundwater_leveL_contours</Name>
    <UserStyle>
      <Name>groundwater_leveL_contours:withContourLabels</Name>
      <Title>withCountourLabels</Title>
      <Abstract>withCountourLabels</Abstract>
      <FeatureTypeStyle>
        <Name>withCountourLabels</Name>
        <Title>withCountourLabels</Title>
<!-- 5 breaks natural jenks, 1.0-6.0, 6.1-15.0, 15.1-35.0, 35.1-55.1, 55.1-80
-->
        <Rule>
          <Name>1.0-6.0</Name>
          <Title>1.0-6.0</Title>
          <Abstract>1.0-6.0</Abstract>
          <ogc:Filter>
            <ogc:PropertyIsLessThanOrEqualTo>
              <ogc:PropertyName>ARC_ELEV</ogc:PropertyName>
              <ogc:Literal>6.0</ogc:Literal>
            </ogc:PropertyIsLessThanOrEqualTo>
```

```
          </ogc:Filter>
          <MinScaleDenominator>0</MinScaleDenominator>
          <MaxScaleDenominator>9999999</MaxScaleDenominator>
          <TextSymbolizer>
            <Label>
              <ogc:PropertyName>ARC_ELEV</ogc:PropertyName>
            </Label>
            <LabelPlacement>
              <LinePlacement />
            </LabelPlacement>
            <Halo>
              <Fill>
                <CssParameter name="fill">#CEF6F5</CssParameter>
                <CssParameter name="fill-opacity">0.6</CssParameter>
              </Fill>
            </Halo>
            <Fill>
              <CssParameter name="fill">#000000</CssParameter>
            </Fill>
            <VendorOption name="followLine">true</VendorOption>
          </TextSymbolizer>
          <LineSymbolizer>
            <Stroke>
              <CssParameter name="stroke">#E3CEF6</CssParameter>
              <CssParameter name="stroke-opacity">0.7</CssParameter>
              <CssParameter name="stroke-width">
                <ogc:Literal>3</ogc:Literal>
              </CssParameter>
            </Stroke>
          </LineSymbolizer>
        </Rule>
<!-- 5 breaks natural jenks, 1.0-6.0, 6.1-15.0, 15.1-35.0, 35.1-55.1, 55.1-80
-->
        <Rule>
          <Name>6.1-15.0</Name>
          <Title>6.1-15.0</Title>
          <Abstract>6.1-15.0</Abstract>
          <ogc:Filter>
            <ogc:And>
              <ogc:PropertyIsGreaterThan>
                <ogc:PropertyName>ARC_ELEV</ogc:PropertyName>
                <ogc:Literal>6.1</ogc:Literal>
              </ogc:PropertyIsGreaterThan>
              <ogc:PropertyIsLessThanOrEqualTo>
                <ogc:PropertyName>ARC_ELEV</ogc:PropertyName>
                <ogc:Literal>15.0</ogc:Literal>
              </ogc:PropertyIsLessThanOrEqualTo>
            </ogc:And>
          </ogc:Filter>
          <MinScaleDenominator>0</MinScaleDenominator>
          <MaxScaleDenominator>9999999</MaxScaleDenominator>
          <TextSymbolizer>
```

```
            <Label>
              <ogc:PropertyName>ARC_ELEV</ogc:PropertyName>
            </Label>
            <LabelPlacement>
              <LinePlacement />
            </LabelPlacement>
            <Halo>
              <Fill>
                <CssParameter name="fill">#CEF6F5</CssParameter>
                <CssParameter name="fill-opacity">0.6</CssParameter>
              </Fill>
            </Halo>
            <Fill>
              <CssParameter name="fill">#000000</CssParameter>
            </Fill>
            <VendorOption name="followLine">true</VendorOption>
          </TextSymbolizer>
          <LineSymbolizer>
            <Stroke>
              <CssParameter name="stroke">#E2A9F3</CssParameter>
              <CssParameter name="stroke-opacity">0.7</CssParameter>
              <CssParameter name="stroke-width">
                <ogc:Literal>3</ogc:Literal>
              </CssParameter>
            </Stroke>
          </LineSymbolizer>
        </Rule>
<!-- 5 breaks natural jenks, 1.0-6.0, 6.1-15.0, 15.1-35.0, 35.1-55.1, 55.1-80
-->
        <Rule>
          <Name>15.1-35.0</Name>
          <Title>15.1-35.0</Title>
          <Abstract>15.1-35.0</Abstract>
          <ogc:Filter>
            <ogc:And>
              <ogc:PropertyIsGreaterThan>
                <ogc:PropertyName>ARC_ELEV</ogc:PropertyName>
                <ogc:Literal>15.1</ogc:Literal>
              </ogc:PropertyIsGreaterThan>
              <ogc:PropertyIsLessThanOrEqualTo>
                <ogc:PropertyName>ARC_ELEV</ogc:PropertyName>
                <ogc:Literal>35.0</ogc:Literal>
              </ogc:PropertyIsLessThanOrEqualTo>
            </ogc:And>
          </ogc:Filter>
          <MinScaleDenominator>0</MinScaleDenominator>
          <MaxScaleDenominator>9999999</MaxScaleDenominator>
          <TextSymbolizer>
            <Label>
              <ogc:PropertyName>ARC_ELEV</ogc:PropertyName>
            </Label>
            <LabelPlacement>
```

```
              <LinePlacement />
            </LabelPlacement>
            <Halo>
              <Fill>
                <CssParameter name="fill">#CEF6F5</CssParameter>
                <CssParameter name="fill-opacity">0.6</CssParameter>
              </Fill>
            </Halo>
            <Fill>
              <CssParameter name="fill">#000000</CssParameter>
            </Fill>
            <VendorOption name="followLine">true</VendorOption>
          </TextSymbolizer>
          <LineSymbolizer>
            <Stroke>
              <CssParameter name="stroke">#F781F3</CssParameter>
              <CssParameter name="stroke-opacity">0.7</CssParameter>
              <CssParameter name="stroke-width">
                <ogc:Literal>3</ogc:Literal>
              </CssParameter>
            </Stroke>
          </LineSymbolizer>
        </Rule>
<!-- 5 breaks natural jenks, 1.0-6.0, 6.1-15.0, 15.1-35.0, 35.1-55.0, 55.1-80
-->
        <Rule>
          <Name>35.1-55.1</Name>
          <Title>35.1-55.1</Title>
          <Abstract>35.1-55.1</Abstract>
          <ogc:Filter>
            <ogc:And>
              <ogc:PropertyIsGreaterThan>
                <ogc:PropertyName>ARC_ELEV</ogc:PropertyName>
                <ogc:Literal>35.1</ogc:Literal>
              </ogc:PropertyIsGreaterThan>
              <ogc:PropertyIsLessThanOrEqualTo>
                <ogc:PropertyName>ARC_ELEV</ogc:PropertyName>
                <ogc:Literal>55</ogc:Literal>
              </ogc:PropertyIsLessThanOrEqualTo>
            </ogc:And>
          </ogc:Filter>
          <MinScaleDenominator>0</MinScaleDenominator>
          <MaxScaleDenominator>9999999</MaxScaleDenominator>
          <TextSymbolizer>
            <Label>
              <ogc:PropertyName>ARC_ELEV</ogc:PropertyName>
            </Label>
            <LabelPlacement>
              <LinePlacement />
            </LabelPlacement>
            <Halo>
              <Fill>
```

```
              <CssParameter name="fill">#CEF6F5</CssParameter>
              <CssParameter name="fill-opacity">0.6</CssParameter>
            </Fill>
          </Halo>
          <Fill>
            <CssParameter name="fill">#000000</CssParameter>
          </Fill>
          <VendorOption name="followLine">true</VendorOption>
        </TextSymbolizer>
        <LineSymbolizer>
          <Stroke>
            <CssParameter name="stroke">#FA58D0</CssParameter>
            <CssParameter name="stroke-opacity">0.7</CssParameter>
            <CssParameter name="stroke-width">
              <ogc:Literal>3</ogc:Literal>
            </CssParameter>
          </Stroke>
        </LineSymbolizer>
      </Rule>
<!-- 5 breaks natural jenks, 1.0-6.0, 6.1-15.0, 15.1-35.0, 35.1-55.1, 55.1-80
-->
      <Rule>
        <Name>55.1-80</Name>
        <Title>55.1-80</Title>
        <Abstract>55.1-80</Abstract>
        <ogc:Filter>
          <ogc:And>
            <ogc:PropertyIsGreaterThan>
              <ogc:PropertyName>ARC_ELEV</ogc:PropertyName>
              <ogc:Literal>55.1</ogc:Literal>
            </ogc:PropertyIsGreaterThan>
            <ogc:PropertyIsLessThanOrEqualTo>
              <ogc:PropertyName>ARC_ELEV</ogc:PropertyName>
              <ogc:Literal>80</ogc:Literal>
            </ogc:PropertyIsLessThanOrEqualTo>
          </ogc:And>
        </ogc:Filter>
        <MinScaleDenominator>0</MinScaleDenominator>
        <MaxScaleDenominator>9999999</MaxScaleDenominator>
        <TextSymbolizer>
          <Label>
            <ogc:PropertyName>ARC_ELEV</ogc:PropertyName>
          </Label>
          <LabelPlacement>
            <LinePlacement />
          </LabelPlacement>
          <Halo>
            <Fill>
              <CssParameter name="fill">#CEF6F5</CssParameter>
              <CssParameter name="fill-opacity">0.6</CssParameter>
            </Fill>
          </Halo>
```

```
            <Fill>
              <CssParameter name="fill">#000000</CssParameter>
            </Fill>
            <VendorOption name="followLine">true</VendorOption>
          </TextSymbolizer>
          <LineSymbolizer>
            <Stroke>
              <CssParameter name="stroke">#FE2E9A</CssParameter>
              <CssParameter name="stroke-opacity">0.7</CssParameter>
              <CssParameter name="stroke-width">
                <ogc:Literal>3</ogc:Literal>
              </CssParameter>
            </Stroke>
          </LineSymbolizer>
        </Rule>
      </FeatureTypeStyle>
    </UserStyle>
  </NamedLayer>
</StyledLayerDescriptor>
```

3. Geology polygons

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<StyledLayerDescriptor                                          version="1.0.0"
xsi:schemaLocation="http://www.opengis.net/sld       StyledLayerDescriptor.xsd"
xmlns="http://www.opengis.net/sld"       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>qmap_clip</Name>
    <UserStyle>
      <Name>qmap_clip:withColours</Name>
      <Title>qmap_clip</Title>
      <Abstract>multiple colours legend</Abstract>
      <!-- FeatureTypeStyles describe how to render different features -->
      <!-- A FeatureTypeStyle for rendering polygons -->
      <FeatureTypeStyle>
        <Name>qmap_clip</Name>
      <!-- STAT_AGE, H, Q, Q1, Q2, Q3, Q4, Q5, Q6, Q8, Tr J, eQ, mQ, water -->
        <Rule>
          <Name>H</Name>
          <Title>H</Title>
          <Abstract>H</Abstract>
          <ogc:Filter>
            <ogc:PropertyIsEqualTo>
              <ogc:PropertyName>STRAT_AGE</ogc:PropertyName>
              <ogc:Literal>H</ogc:Literal>
            </ogc:PropertyIsEqualTo>
          </ogc:Filter>
          <MinScaleDenominator>0</MinScaleDenominator>
          <MaxScaleDenominator>9999999</MaxScaleDenominator>
```

```
        <TextSymbolizer>
          <Label>
            <ogc:PropertyName>MAIN_ROCK</ogc:PropertyName>
          </Label>
          <Font>
            <CssParameter name="font-size">10</CssParameter>
            <CssParameter name="font-color">#000000</CssParameter>
          </Font>
          <LabelPlacement>
            <PointPlacement>
              <AnchorPoint>
                <AnchorPointX>0.5</AnchorPointX>
                <AnchorPointY>0.5</AnchorPointY>
              </AnchorPoint>
            </PointPlacement>
          </LabelPlacement>
          <Fill>
            <CssParameter name="fill">#000000</CssParameter>
          </Fill>

          <VendorOption name="autoWrap">60</VendorOption>
          <VendorOption name="maxDisplacement">150</VendorOption>

        </TextSymbolizer>
        <PolygonSymbolizer>
          <Fill>
            <CssParameter name="fill">#BE81F7</CssParameter>
            <CssParameter name="fill-opacity">0.3</CssParameter>
          </Fill>
          <Stroke>
            <CssParameter name="stroke">#000000</CssParameter>
            <CssParameter name="stroke-width">1</CssParameter>
          </Stroke>
        </PolygonSymbolizer>
      </Rule>
    <!-- STAT_AGE, H, Q, Q1, Q2, Q3, Q4, Q5, Q6, Q8, Tr J, eQ, mQ, water -->
      <Rule>
        <Name>Q</Name>
        <Title>Q</Title>
        <Abstract>Q</Abstract>
        <ogc:Filter>
          <ogc:PropertyIsEqualTo>
            <ogc:PropertyName>STRAT_AGE</ogc:PropertyName>
            <ogc:Literal>Q</ogc:Literal>
          </ogc:PropertyIsEqualTo>
        </ogc:Filter>
        <MinScaleDenominator>0</MinScaleDenominator>
        <MaxScaleDenominator>9999999</MaxScaleDenominator>
        <TextSymbolizer>
          <Label>
            <ogc:PropertyName>MAIN_ROCK</ogc:PropertyName>
          </Label>
```

```
        <Font>
          <CssParameter name="font-size">10</CssParameter>
          <CssParameter name="font-color">#000000</CssParameter>
        </Font>
        <LabelPlacement>
          <PointPlacement>
            <AnchorPoint>
              <AnchorPointX>0.5</AnchorPointX>
              <AnchorPointY>0.5</AnchorPointY>
            </AnchorPoint>
          </PointPlacement>
        </LabelPlacement>
        <Fill>
          <CssParameter name="fill">#000000</CssParameter>
        </Fill>

        <VendorOption name="autoWrap">60</VendorOption>
        <VendorOption name="maxDisplacement">150</VendorOption>

      </TextSymbolizer>
      <PolygonSymbolizer>
        <Fill>
          <CssParameter name="fill">#819FF7</CssParameter>
          <CssParameter name="fill-opacity">0.3</CssParameter>
        </Fill>
        <Stroke>
          <CssParameter name="stroke">#000000</CssParameter>
          <CssParameter name="stroke-width">1</CssParameter>
        </Stroke>
      </PolygonSymbolizer>
    </Rule>
  <!-- STAT_AGE, H, Q, Q1, Q2, Q3, Q4, Q5, Q6, Q8, Tr J, eQ, mQ, water -->
    <Rule>
      <Name>Q1</Name>
      <Title>Q1</Title>
      <Abstract>Q1</Abstract>
      <ogc:Filter>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>STRAT_AGE</ogc:PropertyName>
          <ogc:Literal>Q1</ogc:Literal>
        </ogc:PropertyIsEqualTo>
      </ogc:Filter>
      <MinScaleDenominator>0</MinScaleDenominator>
      <MaxScaleDenominator>9999999</MaxScaleDenominator>
      <TextSymbolizer>
        <Label>
          <ogc:PropertyName>MAIN_ROCK</ogc:PropertyName>
        </Label>
        <Font>
          <CssParameter name="font-size">10</CssParameter>
          <CssParameter name="font-color">#000000</CssParameter>
        </Font>
```

```
          <LabelPlacement>
            <PointPlacement>
              <AnchorPoint>
                <AnchorPointX>0.5</AnchorPointX>
                <AnchorPointY>0.5</AnchorPointY>
              </AnchorPoint>
            </PointPlacement>
          </LabelPlacement>
          <Fill>
            <CssParameter name="fill">#000000</CssParameter>
          </Fill>

          <VendorOption name="autoWrap">60</VendorOption>
          <VendorOption name="maxDisplacement">150</VendorOption>

        </TextSymbolizer>
        <PolygonSymbolizer>
          <Fill>
            <CssParameter name="fill">#81DAF5</CssParameter>
            <CssParameter name="fill-opacity">0.3</CssParameter>
          </Fill>
          <Stroke>
            <CssParameter name="stroke">#000000</CssParameter>
            <CssParameter name="stroke-width">1</CssParameter>
          </Stroke>
        </PolygonSymbolizer>
      </Rule>
    <!-- STAT_AGE, H, Q, Q1, Q2, Q3, Q4, Q5, Q6, Q8, Tr J, eQ, mQ, water -->
      <Rule>
        <Name>Q2</Name>
        <Title>Q2</Title>
        <Abstract>Q2</Abstract>
        <ogc:Filter>
          <ogc:PropertyIsEqualTo>
            <ogc:PropertyName>STRAT_AGE</ogc:PropertyName>
            <ogc:Literal>Q2</ogc:Literal>
          </ogc:PropertyIsEqualTo>
        </ogc:Filter>
        <MinScaleDenominator>0</MinScaleDenominator>
        <MaxScaleDenominator>9999999</MaxScaleDenominator>
        <TextSymbolizer>
          <Label>
            <ogc:PropertyName>MAIN_ROCK</ogc:PropertyName>
          </Label>
          <Font>
            <CssParameter name="font-size">10</CssParameter>
            <CssParameter name="font-color">#000000</CssParameter>
          </Font>
          <LabelPlacement>
            <PointPlacement>
              <AnchorPoint>
                <AnchorPointX>0.5</AnchorPointX>
```

```
            <AnchorPointY>0.5</AnchorPointY>
          </AnchorPoint>
        </PointPlacement>
      </LabelPlacement>
      <Fill>
        <CssParameter name="fill">#000000</CssParameter>
      </Fill>

      <VendorOption name="autoWrap">60</VendorOption>
      <VendorOption name="maxDisplacement">150</VendorOption>

    </TextSymbolizer>
    <PolygonSymbolizer>
      <Fill>
        <CssParameter name="fill">#81F7D8</CssParameter>
        <CssParameter name="fill-opacity">0.3</CssParameter>
      </Fill>
      <Stroke>
        <CssParameter name="stroke">#000000</CssParameter>
        <CssParameter name="stroke-width">1</CssParameter>
      </Stroke>
    </PolygonSymbolizer>
  </Rule>
<!-- STAT_AGE, H, Q, Q1, Q2, Q3, Q4, Q5, Q6, Q8, Tr J, eQ, mQ, water -->
  <Rule>
    <Name>Q3</Name>
    <Title>Q3</Title>
    <Abstract>Q3</Abstract>
    <ogc:Filter>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>STRAT_AGE</ogc:PropertyName>
        <ogc:Literal>Q3</ogc:Literal>
      </ogc:PropertyIsEqualTo>
    </ogc:Filter>
    <MinScaleDenominator>0</MinScaleDenominator>
    <MaxScaleDenominator>9999999</MaxScaleDenominator>
    <TextSymbolizer>
      <Label>
        <ogc:PropertyName>MAIN_ROCK</ogc:PropertyName>
      </Label>
      <Font>
        <CssParameter name="font-size">10</CssParameter>
        <CssParameter name="font-color">#000000</CssParameter>
      </Font>
      <LabelPlacement>
        <PointPlacement>
          <AnchorPoint>
            <AnchorPointX>0.5</AnchorPointX>
            <AnchorPointY>0.5</AnchorPointY>
          </AnchorPoint>
        </PointPlacement>
      </LabelPlacement>
```

```
        <Fill>
          <CssParameter name="fill">#000000</CssParameter>
        </Fill>

        <VendorOption name="autoWrap">60</VendorOption>
        <VendorOption name="maxDisplacement">150</VendorOption>

      </TextSymbolizer>
      <PolygonSymbolizer>
        <Fill>
          <CssParameter name="fill">#81F781</CssParameter>
          <CssParameter name="fill-opacity">0.3</CssParameter>
        </Fill>
        <Stroke>
          <CssParameter name="stroke">#000000</CssParameter>
          <CssParameter name="stroke-width">1</CssParameter>
        </Stroke>
      </PolygonSymbolizer>
    </Rule>
  <!-- STAT_AGE, H, Q, Q1, Q2, Q3, Q4, Q5, Q6, Q8, Tr J, eQ, mQ, water -->
    <Rule>
      <Name>Q4</Name>
      <Title>Q4</Title>
      <Abstract>Q4</Abstract>
      <ogc:Filter>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>STRAT_AGE</ogc:PropertyName>
          <ogc:Literal>Q4</ogc:Literal>
        </ogc:PropertyIsEqualTo>
      </ogc:Filter>
      <MinScaleDenominator>0</MinScaleDenominator>
      <MaxScaleDenominator>9999999</MaxScaleDenominator>
      <TextSymbolizer>
        <Label>
          <ogc:PropertyName>MAIN_ROCK</ogc:PropertyName>
        </Label>
        <Font>
          <CssParameter name="font-size">10</CssParameter>
          <CssParameter name="font-color">#000000</CssParameter>
        </Font>
        <LabelPlacement>
          <PointPlacement>
            <AnchorPoint>
              <AnchorPointX>0.5</AnchorPointX>
              <AnchorPointY>0.5</AnchorPointY>
            </AnchorPoint>
          </PointPlacement>
        </LabelPlacement>
        <Fill>
          <CssParameter name="fill">#000000</CssParameter>
        </Fill>
```

```
            <VendorOption name="autoWrap">60</VendorOption>
            <VendorOption name="maxDisplacement">150</VendorOption>

          </TextSymbolizer>
          <PolygonSymbolizer>
            <Fill>
              <CssParameter name="fill">#BEF781</CssParameter>
              <CssParameter name="fill-opacity">0.3</CssParameter>
            </Fill>
            <Stroke>
              <CssParameter name="stroke">#000000</CssParameter>
              <CssParameter name="stroke-width">1</CssParameter>
            </Stroke>
          </PolygonSymbolizer>
        </Rule>
       <!-- STAT_AGE, H, Q, Q1, Q2, Q3, Q4, Q5, Q6, Q8, Tr J, eQ, mQ, water --
>
        <Rule>
          <Name>Q5</Name>
          <Title>Q5</Title>
          <Abstract>Q5</Abstract>
          <ogc:Filter>
            <ogc:PropertyIsEqualTo>
              <ogc:PropertyName>STRAT_AGE</ogc:PropertyName>
              <ogc:Literal>Q5</ogc:Literal>
            </ogc:PropertyIsEqualTo>
          </ogc:Filter>
          <MinScaleDenominator>0</MinScaleDenominator>
          <MaxScaleDenominator>9999999</MaxScaleDenominator>
          <TextSymbolizer>
            <Label>
              <ogc:PropertyName>MAIN_ROCK</ogc:PropertyName>
            </Label>
            <Font>
              <CssParameter name="font-size">10</CssParameter>
              <CssParameter name="font-color">#000000</CssParameter>
            </Font>
            <LabelPlacement>
              <PointPlacement>
                <AnchorPoint>
                  <AnchorPointX>0.5</AnchorPointX>
                  <AnchorPointY>0.5</AnchorPointY>
                </AnchorPoint>
              </PointPlacement>
            </LabelPlacement>
            <Fill>
              <CssParameter name="fill">#000000</CssParameter>
            </Fill>

            <VendorOption name="autoWrap">60</VendorOption>
            <VendorOption name="maxDisplacement">150</VendorOption>
```

```
        </TextSymbolizer>
        <PolygonSymbolizer>
          <Fill>
            <CssParameter name="fill">#F3F781</CssParameter>
            <CssParameter name="fill-opacity">0.3</CssParameter>
          </Fill>
          <Stroke>
            <CssParameter name="stroke">#000000</CssParameter>
            <CssParameter name="stroke-width">1</CssParameter>
          </Stroke>
        </PolygonSymbolizer>
      </Rule>
    <!-- STAT_AGE, H, Q, Q1, Q2, Q3, Q4, Q5, Q6, Q8, Tr J, eQ, mQ, water -->
      <Rule>
        <Name>Q8</Name>
        <Title>Q8</Title>
        <Abstract>Q8</Abstract>
        <ogc:Filter>
          <ogc:PropertyIsEqualTo>
            <ogc:PropertyName>STRAT_AGE</ogc:PropertyName>
            <ogc:Literal>Q8</ogc:Literal>
          </ogc:PropertyIsEqualTo>
        </ogc:Filter>
        <MinScaleDenominator>0</MinScaleDenominator>
        <MaxScaleDenominator>9999999</MaxScaleDenominator>
        <TextSymbolizer>
          <Label>
            <ogc:PropertyName>MAIN_ROCK</ogc:PropertyName>
          </Label>
          <Font>
            <CssParameter name="font-size">10</CssParameter>
            <CssParameter name="font-color">#000000</CssParameter>
          </Font>
          <LabelPlacement>
            <PointPlacement>
              <AnchorPoint>
                <AnchorPointX>0.5</AnchorPointX>
                <AnchorPointY>0.5</AnchorPointY>
              </AnchorPoint>
            </PointPlacement>
          </LabelPlacement>
          <Fill>
            <CssParameter name="fill">#000000</CssParameter>
          </Fill>

          <VendorOption name="autoWrap">60</VendorOption>
          <VendorOption name="maxDisplacement">150</VendorOption>

        </TextSymbolizer>
        <PolygonSymbolizer>
          <Fill>
            <CssParameter name="fill">#F7BE81</CssParameter>
```

```
        <CssParameter name="fill-opacity">0.3</CssParameter>
      </Fill>
      <Stroke>
        <CssParameter name="stroke">#000000</CssParameter>
        <CssParameter name="stroke-width">1</CssParameter>
      </Stroke>
    </PolygonSymbolizer>
  </Rule>
<!-- STAT_AGE, H, Q, Q1, Q2, Q3, Q4, Q5, Q6, Q8, Tr J, eQ, mQ, water -->
  <Rule>
    <Name>TrJ</Name>
    <Title>Tr J</Title>
    <Abstract>Tr J</Abstract>
    <ogc:Filter>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>STRAT_AGE</ogc:PropertyName>
        <ogc:Literal>Tr J</ogc:Literal>
      </ogc:PropertyIsEqualTo>
    </ogc:Filter>
    <MinScaleDenominator>0</MinScaleDenominator>
    <MaxScaleDenominator>9999999</MaxScaleDenominator>
    <TextSymbolizer>
      <Label>
        <ogc:PropertyName>MAIN_ROCK</ogc:PropertyName>
      </Label>
      <Font>
        <CssParameter name="font-size">10</CssParameter>
        <CssParameter name="font-color">#000000</CssParameter>
      </Font>
      <LabelPlacement>
        <PointPlacement>
          <AnchorPoint>
            <AnchorPointX>0.5</AnchorPointX>
            <AnchorPointY>0.5</AnchorPointY>
          </AnchorPoint>
        </PointPlacement>
      </LabelPlacement>
      <Fill>
        <CssParameter name="fill">#000000</CssParameter>
      </Fill>

      <VendorOption name="autoWrap">60</VendorOption>
      <VendorOption name="maxDisplacement">150</VendorOption>

    </TextSymbolizer>
    <PolygonSymbolizer>
      <Fill>
        <CssParameter name="fill">#F78181</CssParameter>
        <CssParameter name="fill-opacity">0.3</CssParameter>
      </Fill>
      <Stroke>
        <CssParameter name="stroke">#000000</CssParameter>
```

```
                <CssParameter name="stroke-width">1</CssParameter>
              </Stroke>
            </PolygonSymbolizer>
          </Rule>
                <!-- STAT_AGE, H, Q, Q1, Q2, Q3, Q4, Q5, Q6, Q8, Tr J, eQ, mQ,
water -->
          <Rule>
            <Name>eQ</Name>
            <Title>eQ</Title>
            <Abstract>eQ</Abstract>
            <ogc:Filter>
              <ogc:PropertyIsEqualTo>
                <ogc:PropertyName>STRAT_AGE</ogc:PropertyName>
                <ogc:Literal>eQ</ogc:Literal>
              </ogc:PropertyIsEqualTo>
            </ogc:Filter>
            <MinScaleDenominator>0</MinScaleDenominator>
            <MaxScaleDenominator>9999999</MaxScaleDenominator>
            <TextSymbolizer>
              <Label>
                <ogc:PropertyName>MAIN_ROCK</ogc:PropertyName>
              </Label>
              <Font>
                <CssParameter name="font-size">10</CssParameter>
                <CssParameter name="font-color">#000000</CssParameter>
              </Font>
              <LabelPlacement>
                <PointPlacement>
                  <AnchorPoint>
                    <AnchorPointX>0.5</AnchorPointX>
                    <AnchorPointY>0.5</AnchorPointY>
                  </AnchorPoint>
                </PointPlacement>
              </LabelPlacement>
              <Fill>
                <CssParameter name="fill">#000000</CssParameter>
              </Fill>

              <VendorOption name="autoWrap">60</VendorOption>
              <VendorOption name="maxDisplacement">150</VendorOption>

            </TextSymbolizer>
            <PolygonSymbolizer>
              <Fill>
                <CssParameter name="fill">#A4A4A4</CssParameter>
                <CssParameter name="fill-opacity">0.3</CssParameter>
              </Fill>
              <Stroke>
                <CssParameter name="stroke">#000000</CssParameter>
                <CssParameter name="stroke-width">1</CssParameter>
              </Stroke>
            </PolygonSymbolizer>
```

```
        </Rule>
              <!-- STAT_AGE, H, Q, Q1, Q2, Q3, Q4, Q5, Q6, Q8, Tr J, eQ, mQ,
water -->
        <Rule>
          <Name>mQ</Name>
          <Title>mQ</Title>
          <Abstract>mQ</Abstract>
          <ogc:Filter>
            <ogc:PropertyIsEqualTo>
              <ogc:PropertyName>STRAT_AGE</ogc:PropertyName>
              <ogc:Literal>mQ</ogc:Literal>
            </ogc:PropertyIsEqualTo>
          </ogc:Filter>
          <MinScaleDenominator>0</MinScaleDenominator>
          <MaxScaleDenominator>9999999</MaxScaleDenominator>
          <TextSymbolizer>
            <Label>
              <ogc:PropertyName>MAIN_ROCK</ogc:PropertyName>
            </Label>
            <Font>
              <CssParameter name="font-size">10</CssParameter>
              <CssParameter name="font-color">#000000</CssParameter>
            </Font>
            <LabelPlacement>
              <PointPlacement>
                <AnchorPoint>
                  <AnchorPointX>0.5</AnchorPointX>
                  <AnchorPointY>0.5</AnchorPointY>
                </AnchorPoint>
              </PointPlacement>
            </LabelPlacement>
            <Fill>
              <CssParameter name="fill">#000000</CssParameter>
            </Fill>

            <VendorOption name="autoWrap">60</VendorOption>
            <VendorOption name="maxDisplacement">150</VendorOption>

          </TextSymbolizer>
          <PolygonSymbolizer>
            <Fill>
              <CssParameter name="fill">#FA8258</CssParameter>
              <CssParameter name="fill-opacity">0.3</CssParameter>
            </Fill>
            <Stroke>
              <CssParameter name="stroke">#000000</CssParameter>
              <CssParameter name="stroke-width">1</CssParameter>
            </Stroke>
          </PolygonSymbolizer>
        </Rule>
              <!-- STAT_AGE, H, Q, Q1, Q2, Q3, Q4, Q5, Q6, Q8, Tr J, eQ, mQ,
water -->
```

```
        <Rule>
          <Name>Water </Name>
          <Title>Water </Title>
          <Abstract>Water </Abstract>
          <ogc:Filter>
            <ogc:PropertyIsEqualTo>
              <ogc:PropertyName>STRAT_AGE</ogc:PropertyName>
              <ogc:Literal></ogc:Literal>
            </ogc:PropertyIsEqualTo>
          </ogc:Filter>
          <MinScaleDenominator>0</MinScaleDenominator>
          <MaxScaleDenominator>9999999</MaxScaleDenominator>
          <TextSymbolizer>
            <Label>
              <ogc:PropertyName>MAIN_ROCK</ogc:PropertyName>
            </Label>
            <Font>
              <CssParameter name="font-size">10</CssParameter>
              <CssParameter name="font-color">#000000</CssParameter>
            </Font>
            <LabelPlacement>
              <PointPlacement>
                <AnchorPoint>
                  <AnchorPointX>0.5</AnchorPointX>
                  <AnchorPointY>0.5</AnchorPointY>
                </AnchorPoint>
              </PointPlacement>
            </LabelPlacement>
            <Fill>
              <CssParameter name="fill">#000000</CssParameter>
            </Fill>

            <VendorOption name="autoWrap">60</VendorOption>
            <VendorOption name="maxDisplacement">150</VendorOption>

          </TextSymbolizer>
          <PolygonSymbolizer>
            <Fill>
              <CssParameter name="fill">#5882FA</CssParameter>
              <CssParameter name="fill-opacity">0.3</CssParameter>
            </Fill>
            <Stroke>
              <CssParameter name="stroke">#000000</CssParameter>
              <CssParameter name="stroke-width">1</CssParameter>
            </Stroke>
          </PolygonSymbolizer>
        </Rule>
      </FeatureTypeStyle>
    </UserStyle>
  </NamedLayer>
</StyledLayerDescriptor>
```

(all customised styles provided on DVD)

B.4 Web server installation and configuration

1. Apt-get packages, command line and configuration summary

```
#> apt-get update
#> apt-get install apache2
#> apt-get install openjdk-7-jdk (for geoserver etc)
#> apt-get install postgresql-9.1-postgis postgis postgresql-9.1
#> useradd -m geoserver (dedicated non-root-user for geoserver instance)
#> update-alternatives --config java (because I installed openjdk7)
#> vi /etc/apache2/sites-enabled/000-default (password protection for complete
site)
#> htpasswd -c /home/geoserver/includes/htpasswd sbg (http password for sbg
user)
#> a2enmod proxy auth_basic authn_file proxy_http rewrite
-   configured proxying and password security for webserver in
    /etc/apache2/sites-enabled/default
-   configured tomcat security in /home/geoserver/tomcat6/conf/tomcat-users.xml
    (role manager-gui)
-   configured tomcat for localhost only in
    /home/geoserver/tomcat6/conf/server.xml (address=127.0.0.1)
-   deployed geoserver.war in /home/geoserver/tomcat6/webapps

#> apt-get install proftpd-basic
-   edited config file /etc/proftpd/proftpd.conf for virtual users under /vol1
    and shared identity for easy file maintenance
-   ftp users and passwords in /vol1/conf


#> apt-get install libapache2-mod-php5 php5-pgsql  (for web php based postgre
sql admin console)
-   enabled php in apache2
-   downloaded PhpPgAdmin, edited apache2 config again to include it
-   configured includes/phppgadmin/conf/conig.inc.php for localhost
```

2. Postgresql/PostGIS plus tables

```
#> su postgres -c psql template1
psql>  ALTER USER postgres WITH PASSWORD 'TheSecr3tP@ss';
psql>  CREATE DATABASE template_postgis WITH TEMPLATE = template1 ENCODING =
'UTF8';
psql>  \i /usr/share/postgresql/9.1/contrib/postgis-1.5/postgis.sql
psql>  \i /usr/share/postgresql/9.1/contrib/postgis-1.5/spatial_ref_sys.sql
psql>  UPDATE pg_database set datistemplate = TRUE where datname =
'template_postgis';
psql>  grant all on geometry_columns to public;
psql>  grant all on spatial_ref_sys to PUBLIC ;
```

```
psql>  create database my_geoserver_data_store with template =
template_postgis;
psql>  \q
#>
#> passwd -d postgres
#> su postgres -c passwd
#>
```

(full installation packages/server image on DVD)

References

Atkinson, R., Dornblut, I. and Smith, D., 2012, An international standard conceptual model for sharing references to hydrologic features. Journal of Hydrology, 424–425: 24-36.

Australia, C.o., 1999, The Australian National Groundwater Data Transfer Standard (NLP Project 97/7020). Working Group on National Groundwater Data Standards, http://www.brs.gov.au/land&water/groundwater/xcel/gw_std.doc.

Beard, D., Year, Using VRML to share large volumes of complex 3D geoscientific information via the Web. Web3D 2006 Proceedings of the eleventh international conference on 3D web technology, Columbia, Maryland, 163-167.

Behr, J., Eschler, P., Jung, Y. and Zöllner, M., Year, X3DOM: a DOM-based HTML5/X3D integration model. Web3D 2009 Proceedings of the 14th International Conference on 3D Web Technology, Darmstadt, Germany, 127-135.

Boisvert, É. and Brodaric, B., 2011, GWML 1.1 - GML Application Schema API Specification. National Resources Canada, http://ngwd-bdnes.cits.rncan.gc.ca/service/api_ngwds/en/gwml.html.

Breunig, M. and Zlatanova, S., 2011, 3D geo-database research: Retrospective and future directions. Computers & Geosciences, 37: 791-803.

Brodaric, B. and Booth, N., 2011, OGC® Groundwater Interoperability Experiment - FINAL REPORT, Report, Open Geospatial Consortium, Inc.

Bröring, A., Echterhoff, J., Jirka, S., Simonis, I., Everding, T., Stasch, C., Liang, S. and Lemmens, R., 2011, New Generation Sensor Web Enablement. Sensors, 11: 2652-2699.

Chang, Y.-S. and Park, H.-D., 2004, Development of a web-based Geographic Information System for the management of borehole and geological data. Computers & Geosciences, 30: 887–897.

Daughney, D.J. and Reeves, R.R., 2006, Analysis of temporal trends in New Zealand's groundwater quality based on data from the National Groundwater Monitoring Programme. Journal of Hydrology (NZ), 45: 4-62.

Directive, 2007/2/EC, Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE).

Fils, D., Cervato, C., Reed, J., Diver, P., Tang, X., Bohling, G. and Greer, D., 2009, CHRONOS architecture: Experiences with an open-source services-oriented architecture for geoinformatics. Computers & Geosciences, 35: 774-782.

Giuliani, G., Ray, N. and Lehmann, A., 2011, Grid-enabled Spatial Data Infrastructure for environmental sciences: Challenges and opportunities. Future Generation Computer Systems, 27: 292-303.

Hildebrandt, D. and Döllner, J., 2010, Service-oriented, standards-based 3D geovisualization: Potential and challenges. Computers, Environment and Urban Systems, 34: 484-495.

Huang, M., Maidment, D.R. and Tian, Y., 2011, Using SOA and RIAs for water data discovery and retrieval. Environ. Model. Softw., 26: 1309-1324.

ISO/IEC, 1997, Virtual Reality Modeling Language - ISO/IEC 14772-1:1997 ISO/IEC, http://www.web3d.org/x3d/specifications/vrml/ISO-IEC-14772-VRML97/.

ISO/IEC, 2008, Extensible 3D (X3D) - ISO/IEC 19775-1.2:2008. ISO/IEC, http://www.web3d.org/x3d/specifications/.

Kessler, H., Mathers, S. and Sobisch, H.-G., 2009, The capture and dissemination of integrated 3D geospatial knowledge at the British Geological Survey using GSI3D software and methodology. Computers & Geosciences, 35: 1311-1321.

Klug, H., Daughney, C., Verhagen, F., Westerhoff, R. and Ward, N.D., 2011, Freshwater resources management: Starting SMART characterization of New Zealand's aquifers, http://www.earthzine.org/2011/12/13/freshwater-resources-management-starting-smart-characterization-of-new-zealands-aquifers/.

Kmoch, A., Klug, H. and White, P., Year, Freshwater resources management: first steps towards the characterisation of New Zealand's aquifers. GI_Forum 2012: Geovisualization, Society and Learning, Salzburg, 2012, 376-385.

LINZ, 2012, Land Information New Zealand. LINZ, http://www.linz.govt.nz.

MFE, 2012, Ministry of the Environment New Zealand. MFE, http://www.mfe.govt.nz.

Mittlböck, M., Morper-Busch, L., Atzl, C. and Klug, H., Year, Task-orientierte Web-Maps zur kompakten Visualisierung kartographischer Inhalte Angewandte Geoinformatik 2012, Salzburg, 333-338.

MSI, 2012, Ministry of Science and Innovation New Zealand. MSI, http://www.msi.govt.nz.

OGC, 2006, OpenGIS Web Map Service (ISO 19128) WMS v1.3.0 The Open Geospatial Consortium (OGC), http://www.opengeospatial.org/standards/is.

OGC, 2007a, OpenGIS® Web Processing Service WPS v1.0.0, 2007-06-08, OGC 05-007r7, Peter Schut. The Open Geospatial Consortium (OGC), http://www.opengeospatial.org/standards/is.

OGC, 2007b, OpenGIS Catalogue Service Implementation Specification (ISO 19115), v2.0.2. The Open Geospatial Consortium (OGC), http://www.opengeospatial.org/standards/is.

OGC, 2007c, OpenGIS® Geography Markup Language (GML) Encoding Standard (ISO 19136), 2007-08-27, OGC 07-036, Version: 3.2.1, Clemens Portele. The Open Geospatial Consortium (OGC), http://www.opengeospatial.org/standards/is.

OGC, 2007d, Styled Layer Descriptor profile of the Web Map Service Implementation Specification, 2007-06-29, OGC 05-078r4, Version: 1.1.0, Dr. Markus Lupp. The Open Geospatial Consortium (OGC), http://www.opengeospatial.org/standards/is.

OGC, 2010a, OGC WCS 2.0 Interface Standard - Core, v2.0. The Open Geospatial Consortium (OGC), http://www.opengeospatial.org/standards/is.

OGC, 2010b, OpenGIS Web Feature Service 2.0 (ISO 19142) WFS v2.0. The Open Geospatial Consortium (OGC), http://www.opengeospatial.org/standards/is.

OGC, 2010c, Draft for Candidate OpenGIS Web 3D Service Interface Standard. The Open Geospatial Consortium (OGC), http://www.opengeospatial.org/standards/is.

OGC, 2011a, Observations and Measurements - XML Implementation, O&M v2.0. The Open Geospatial Consortium (OGC), http://www.opengeospatial.org/standards/is.

OGC, 2011b, OGC SWE Common Data Model Encoding Standard, v2.0. The Open Geospatial Consortium (OGC), http://www.opengeospatial.org/standards/is.

OGC, 2011c, OpenGIS® Implementation Standard for Geographic information - Simple feature access, 2011-05-28, OGC 06-103r4, John R. Herring. The Open Geospatial Consortium (OGC), http://www.opengeospatial.org/standards/is.

OGC, 2012a, OGC Sensor Observation Service Interface Standard, v2.0. The Open Geospatial Consortium (OGC), http://www.opengeospatial.org/standards/is.

OGC, 2012b, OGC Standards. Open Geospatial Consortium, http://www.opengeospatial.org/standards.

Pebesma, E., Cornford, D., Dubois, G., Heuvelink, G.B.M., Hristopulos, D., Pilz, J., Stöhlker, U., Morin, G. and Skøien, J.O., 2011, INTAMAP: The design and implementation of an interoperable automated interpolation web service. Computers & Geosciences, 37: 343-352.

Reitz, T., Krämer, M. and Thum, S., 2009, A processing pipeline for X3D earth-based spatial data view services. ACM, Darmstadt, Germany.

SMART, 2012, SMART aquifer characterisation project homepage. SMART consortium, http://www.smart-project.info.

Smolders, S., Alegre, C., Gianfranceschi, S., Gilles, M., Resch, B. and Everding, T., 2011, GENESIS Employing Web Processing Services and Sensor Web Technology for Environmental Management, Utrecht, (NL).

White, P.A. and Reeves, R.R., 1999, Waimea Plains aquifer structure as determined by threedimensional computer modelling. J. Hydrol. NZ, 40: 49-75.

White, P.A., 2006, Some Future Directions in Hydrology. Journal of Hydrology (NZ), 45: 63-68.

White, P.A., Tschritter, C. and Cameron, S., 2011, A 3D Geological Model of the Upper Waikato River Catchment with Associated Hydrogeological Parameters, Wellington.

X3DOM, 2012, X3DOM - an experimental open source framework and runtime to integrate HTML5 and declarative 3D content.

Statutory Declaration

I hereby declare that this thesis has not been submitted to obtain an academic degree at any other university. All work is genuine and carried out by myself unless stated otherwise in the acknowledgments. I authorise the University of Salzburg to lend this document, or reproductions of this document by photocopying or any other means, in total or in part, at the request of other institutions or individuals for the purpose of research. The copyright remains with the author or with the respective media, in which one of the articles has been published. No quotation from this thesis may be published unless the source is properly acknowledged. All graphics and illustrations were designed and drawn by the author if not stated differently.

____________________________ ____________________________

Place, Date Signature