

Building SDIs with Free Software – the deegree project

Jens FITZKE & Klaus GREVE & Markus MÜLLER & Andreas POTH

lat-lon – raumbezogene Informationssysteme, Deutschland Meckenheimer Allee 176, 53115 Bonn, Germany, <http://www.lat-lon.de>
{fitzke|greve|mueller|poth}@lat-lon.de

1 INTRODUCTION

Building Spatial Data Infrastructures (SDI) raises organizational as well as technical challenges. On the technical side, the use of standardized components guarantees sustainability of investments due to a vendor-neutral architecture. The SDI acts as a whole, while its parts be built using software from different vendors. The Free Software project¹ deegree started as a framework for integrating software products of different vendors on base of OpenGIS® and ISO standards². After three years of continuous development it includes now all components necessary for building SDIs. Its Free Software nature opens new ways of cooperation between software vendors/consultants and the organizations responsible for building SDIs. Available resources can be used where they are needed most, and financial funding can be substituted by a collaborative work environment. Compared to other strategies it is easier to start SDI activities on a low cost basis. And it is easier to spread SDI infrastructure from a nucleus to many users without managing (and balancing) complex and precious license agreements.

In 2000 the GIS division of the Geography Department of Bonn University together with its spin-off company lat/lon started to develop a software library called JaGO – Java framework for Geographical Solutions. It was planned to be used in a variety of projects concerned with spatial data processing and should be based as intensively as possible on OpenGIS® standards. Its predecessor was a project called "Simple Features for Java", a project developing a java library for geodata processing based on the Simple Features concept. Learning from this approach, the first part of JaGO was a bunch of java interfaces and classes for the modelling of geodata based on ISO 19107 that allows a more general approach than Simple Features.

In the meantime the project changed its name to deegree and now includes implementations of the greater number of OGC Web Services needed for building SDIs, namely Web Map Service, Web Feature Service, Web Coverage Service, Gazetteer Service, Catalog Service, Web Coordinate Transformation Service and Web Terrain Service. Together with two other Free Software projects deegree even provides the official Reference Implementations for WMS and WFS.

deegree is used in a number of SDI projects over the world, e.g. its catalog service in the national German SDI portal "GeoMIS.Bund" and in the Northrhine-Westfalian state SDI, its WMS and WFS as part of the City of Hamburg's local SDI and the national SDI of the Grand Duchy of Luxembourg, and last but not least its Gazetteer Service on top of the USGS Geographical Names Information System.

a) deegree Architecture overview

The deegree architecture is based on the service paradigm – and more specifically on the geoprocessing concepts of the OGC and ISO/TC 211. From a conceptual point of view the Portrayal Model (Cuthbert 1998) helps to outline the deegree service architecture. The model describes the map production process with four processing units and four representation components (Figure 1). A data selection process which is controlled by query constraints delivers Features – the key concept of OGC/ISO-standardised geoprocessing (OGC 1999). The selection interface technology depends on the DCP type (distributed computing platform). Access to vector data via CORBA, SQL, or OLE/COM is provided by implementations of the Simple Features Specification, whereas the corresponding technology for the Web is defined by the Web Feature Service Specification (WFS). A typical representation for Features and Feature Collections is a GML (Geography Markup Language) Dataset.³

A Display Element Generator applies style rules to features and produces a graphical representation. The resulting Display Elements can be rendered into an image by using constraints such as color depth, image size, and image format. Implementations of the OGC Web Map Service Specification (WMS) which defines an interface to access map images thus represent both the Display Elements Generator and Render process components. The resulting map image finally is displayed to the user. Today, a typical display component for map images in a Web environment is a standard Web browser.

In deegree, each of the processing, representation, and constraints components is represented by Java interfaces or classes. Each deegree Web service implements the interface `org.deegree.services.OCGWebService`. Accordingly, the WMS implementation is realized in the class `org.deegree_impl.services.wms.WMSservice`. The `OCGWebService` interface simply defines one method - `doService(...)` - that receives an `OCGWebServiceEvent` object and starts the process of handling the WMS request (`GetCapabilities`, `GetMap`, `GetFeatureInfo`). The submitted `OCGWebServiceEvent` contains the request to perform and information concerning the class which shall receive the result.

The `WMSservice` class then delegates the performing of the request to a:

1. `MapFilterHandler` for accessing the required data
2. `DEGHandler` for generating display elements from the geospatial data and the associated portrayal rules
3. `RendererHandler` for rendering the display elements to a map

¹Read more about Free Software in general on the Free Software Foundation's Web Site: <http://www.gnu.org/>. "Open Source Software" is a competing concept of the Open Source Initiative with more emphasis on practical than on social aspects (<http://www.opensource.org/>).

²See <http://www.opengis.org/> and <http://www.isotc211.org/>

³All adopted OpenGIS® specifications are available on the OGC Web Site: <http://www.opengis.org/specs/?page=specs>

The WMS is a servlet-based implementation (org.deegree_impl.enterprise.WMSServlet) which acts as a client to WMServices objects and offers the network connection interfaces.

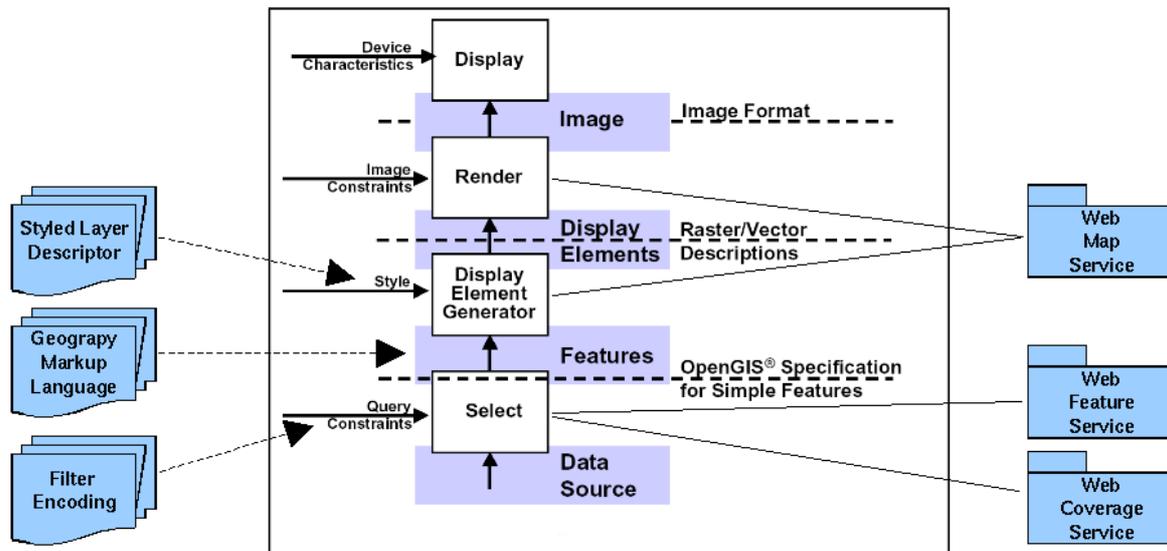


Figure 1: The Portrayal Model (derived from Cuthbert 1998)

1.1.1 Data Access

In the sense of the Portrayal Model a WMS uses a filter service to access the data needed to render the requested map and/or to perform GetFeatureInfo requests. For flexible data access the deegree WMS doesn't implement specific classes for accessing data but offers access to Data sources as WFS or WCS (org.deegree.services.filterservice package). One WFS or WCS can be registered as datasource to the WMS. This does not necessarily mean that an actual Web Service has to be accessed but that data sources (even Java Classes) act like WFS or WCS (Figure 2).

The required additional configuration information is added to the configuration (capabilities) document of the WMS and will be provided by implementations of the interfaces contained in the org.deegree.services.wms.capabilities packages. Because only a small amount of additional information is needed no extra configuration package exists. If a client sends a GetCapabilities request the additional information will be filtered out of the capabilities before the document is sent to the client.

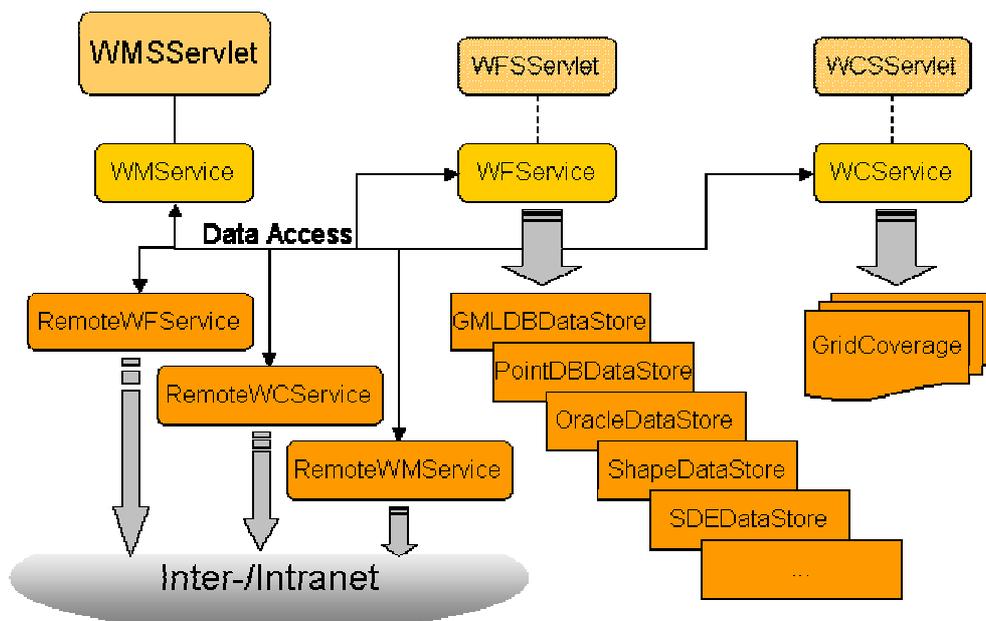


Figure 2: Data access in deegree WMS (from deegree WMS documentation⁴)

A typical internal service chain invoked by a GetMap request to the WMS is outlined in Figure 3. Here, the Web Coordinate Transformation Service (WCTS) is used to perform a coordinate transformation.

⁴<http://deegree.sourceforge.net/>

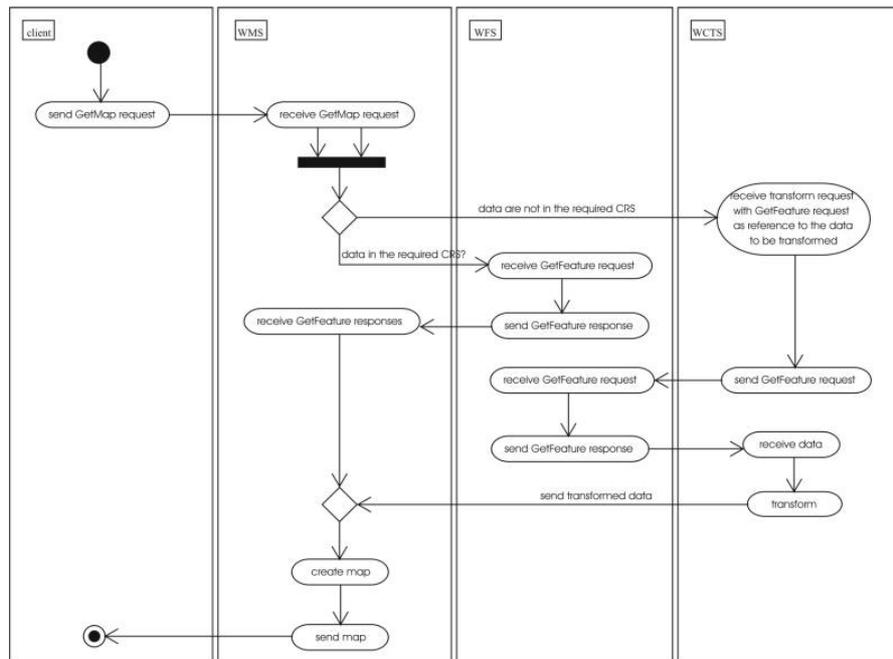


Figure 3: UML activity diagram for a WMS request

1.1.2 From Services to Spatial Data Infrastructures

As pointed out in the GSDI cookbook (Nebert 1991) ...

The word infrastructure is used to promote the concept of a reliable, supporting environment, analogous to a road or telecommunications network, that, in this case, facilitates the access to geographically-related information using a minimum set of standard practices, protocols, and specifications.

From a technical point of view "standard" is the key property of SDI components allowing for plug-and-play interoperability of geospatial information resources. Today, Web Service technology provides a stable baseline for network-based collaboration through all application domains. The OGC, in close collaboration with ISO/TC 211, is providing specifications for a variety of tasks or components necessary for Spatial Data Infrastructures based on concepts which anticipate functionality of the Semantic Web, the intended successor of the WWW.

In deegree the following OGC-based services are available:

Name	Functionality
Web Map Service (WMS)	Web-based creation of maps out of raster and vector datasets. The generated maps can be visualized by common web browsers.
Web Feature Service (WFS)	Web-based access to vector geo-data, that is delivered as GML 2.1.1 conformant XML-documents to clients, which can further process this data (for example in a desktop GIS).
Web Coverage Service (WCS)	Web-based access to raster geo-data, that can be delivered in several image formats (e.g. TIFF, GIF, JPEG, BMP, PNM) and can then further be processed.
Web Catalog Service (CS-W) based on OGC Web Services Stateless Catalog Profile.	Web-based catalog service for administration and querying of metadata describing geo-data and geo-services. A catalog service allows retrieval of data and services based on spatial and textual search criteria.
Web Gazetteer Service (WFS-G)	Service allowing geo-referencing of geographic entities based on textual identifiers (e.g. place names)
Web Terrain Service (WTS)	Creation of views out of 3D-data like city models and digital elevation models. The generated views can be visualized by standard web browsers.
Web Coordinate Transformation Service (WCTS)	A WCTS allows the web-based transformation of geographic coordinates from one coordinate references system into another.

The collaboration of these services is shown in Figure 4.

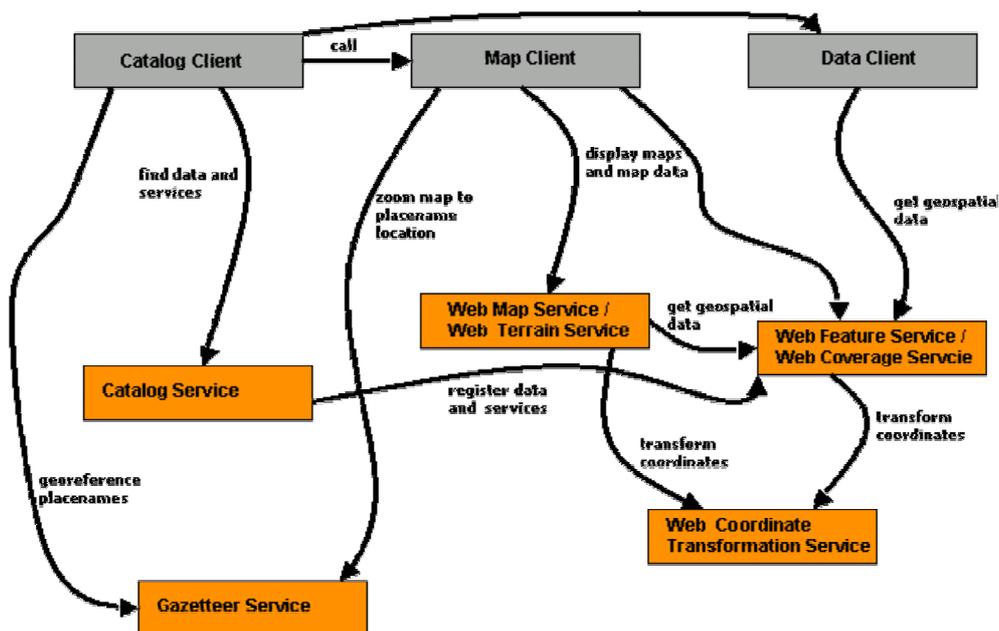


Figure 4: Collaboration of OpenGIS® Web Services

The creation of an SDI with degree Web Services can be realized in two (or more) stages.

1.1.3 Stage 1: Catalog and Map Service

In this first stage a WMS and a CS-W is being installed as a base system allowing querying metadata, finding geodata and displaying it in a intra/internet (Figure 5). This can be coupled with an access control component, allowing only limited access to maps and metadata.

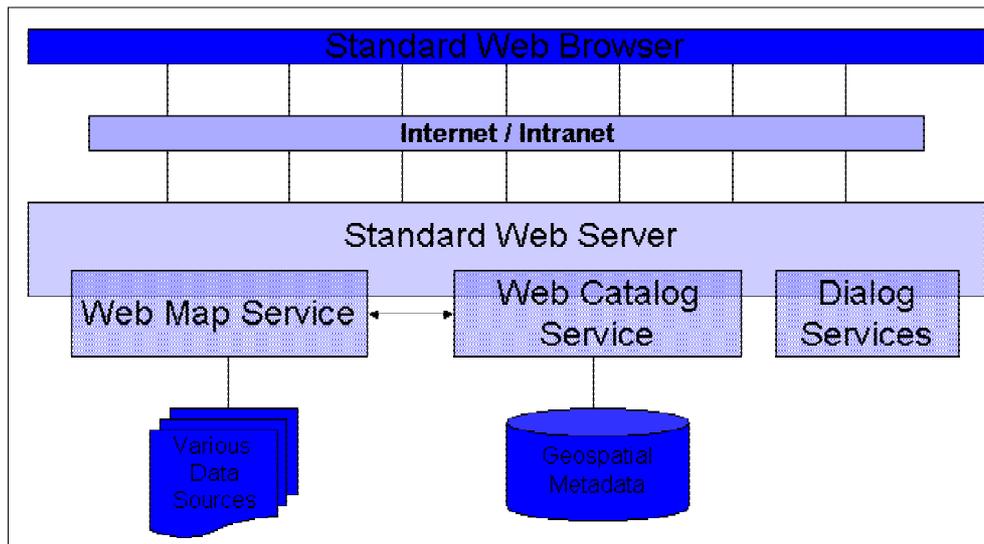
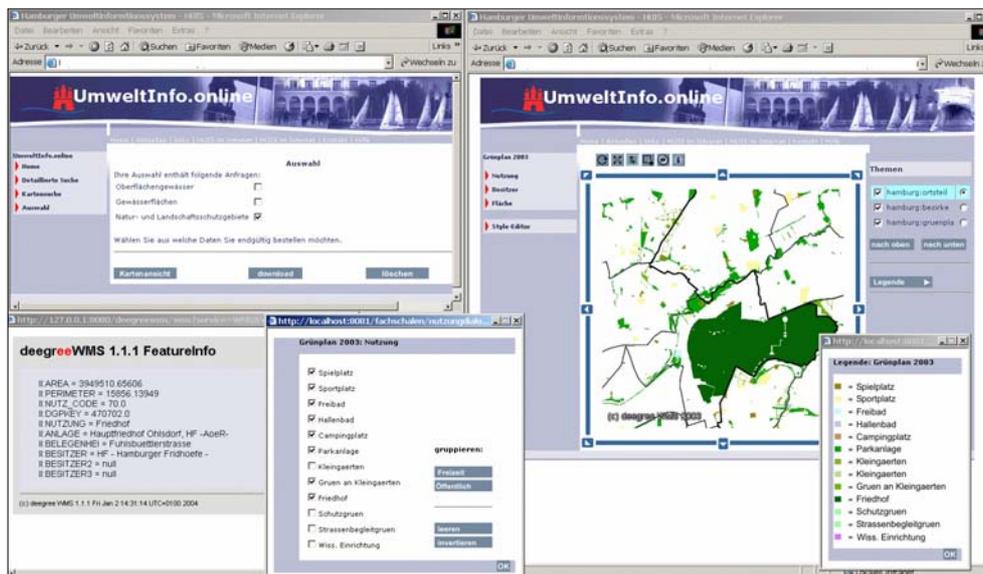


Figure 5: A simple SDI base containing a Map Service and a Catalog Service

The metadata catalog stores its data conforming to ISO 19115 (geodata metadata) and 19119 (services metadata). With the WFS-based implementation of the degree Catalog Service object-relational mapping capabilities can be used to convert legacy metadata schemas on-the-fly to ISO 19115.

Figure 6 shows a compilation of user interface components provided by the application server components provided within stage 1. These are called "Dialog Services" in Figure 5. The upper-left browser window shows the shopping basket interface, which contains selected datasets from one or more catalog queries. From here, the user can either view or download the data depending on his/her privileges – supposed that an access control component is integrated into the application server. The "view dataset" button switches to the WMS client component shown in the upper-right browser window. The popup-windows show dependent components of the WMS client with: 1) information on a selected feature, 2) layer control functionalities, and 3) a map legend (from lower-left to lower-



right).

Figure 6: Screenshot from the user interface of the environmental data infrastructure "UmweltInfo.online" for the city of Hamburg, Germany (cf. Müller et al. 2003)

(a) Stage 2: Direct access to geodata

In the second stage the architecture is extended by further services. By integrating WFS and WCS, more than visualization is possible. Geodata can be directly accessed, processed and adapted data may be made available by persons with proper access rights. Especially for the manipulation of geodata, but also for complex data retrieval, specialized clients have to be installed. These also allow for creation (digitization) of geodata. Gazetteer and Coordinate Transformation service can be included to support querying and data supplying capabilities.

Because the second stage only extends the existing architecture by integrating further components, the overall architecture does not have to be changed. The Spatial Application Server now supports client-functionalities for further services and allows use of these by

way of an easy to use and highly customizable user interface. Furthermore, access control is handled by this SDI layer since a corresponding and adopted concept does not yet exist in Web Services-based geoprocessing.

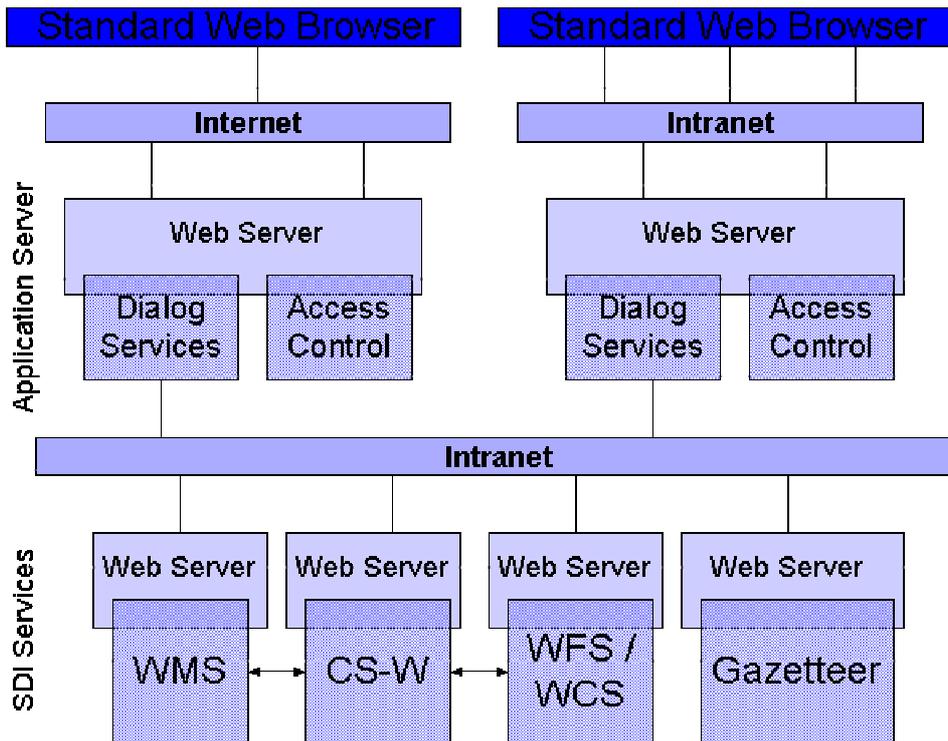


Figure 7: A more elaborated and distributed SDI, including Catalog, Map View and Data Access components

WMS, WFS, WCS, WCAS and Gazetteer interact in controlling data access cooperatively. WMS and Gazetteer support use of the Catalog Service in providing mechanisms for spatial retrieval and/or using geographic names. The Catalog Service identifies relevant datasets of a query and stores information allowing access to the data over suitable Web Services (WMS, WCS, WFS) or by ordering digital copies of the data on CD or DVD.

All services can be realized decentralized. They can reside on different servers, maintained by different organizations. This allows decentralized data storage by the responsible parties for the specific data holdings. Besides, the system is easily scalable as services can be moved to dedicated servers in case this becomes necessary.

Why building SDIs with Free Software?

It is "the double openness" which makes free and interoperable Software so effective in SDI build-up. As far as only the interoperability aspect is considered there is no difference between free and proprietary software, both implementing standardised interfaces.

The freedom⁵ aspect gives even more options. In general, the notion of a world-wide user and developer community is a great advantage: in the majority of cases problems reported to the community are solved promptly. The same is true for a local development community consisting of team members of different organisations, e.g. IT staff from the SDI-driving corporate body together with software engineering staff from a GIS company.

This kind of code level-based openness or collaboration is needed whenever the technical or user requirements exceed the functionality of a software package. Two real-world problems may exemplify this issue: The implementation of thick clients and the integration of exotic spatial reference systems.

As already mentioned, the requirements for clients are increasing when the system becomes more elaborate or more complex queries or transactions are executed. Depending on the profile of the user (simple user up to administrator privileges), different clients can be utilized. A standard user might be allowed to query and visualize specific datasets, for which a HTML-based thin client may be sufficient. In this case the "thin client" only includes the Display component of the Portrayal Model (Figure 1) whereas the underlying services are part of the "fat server". If a user – supposed that he/she has the privileges to do this – wants to be able to edit data in his/her client a "fat client" is needed. This application may then comprise the three upper processing components, including the Feature representation layer of the Portrayal Model whereas the corresponding server simply acts as a Select component. The implementation of this scenario asks for software assembly on at least API level. Full access to the source code improves the performance of the development process a lot and may even help to solve problems which exceed the API level.

⁵Free Software is defined by four kinds of freedom: The freedom to ... 1) run the program, for any purpose – 2) study how the program works, and adapt it to your needs (access to the source code!) – 3) redistribute copies so you can help your neighbor – 4) improve the program, and release your improvements to the public, so that the whole community benefits. (<http://www.gnu.org/philosophy/free-sw.html>)

The support of exotic spatial reference systems is a second example for real-world problems easily solved by a free geoprocessing software package. A lot of spatial reference systems exist which are used only in a small area or within a small group of institutions. Access to data referenced to these systems can only be realized by adding transformation capabilities to the respective software. Again, this task can more easily be accomplished with full access to the source code.

Knowing about these advantages and willing to provide a deep insight into a technical solution the Open GIS Consortium has decided to rely on free software in its reference implementations for the various services:

An excellent way of providing reference implementations is via open source software under the GNU Public License, since the code can be examined in full without limiting the ability of companies to develop commercial products. For these reasons, an open source Web Feature server and client, and a Web Map server were developed for the CITE initiative [...]

(OGC 2003)

The OGC reference implementations support the conformance testing program in providing components for the online test suites and in gaining access to the internals of a tested implementations. In the above named CITE initiative⁶ the free software projects deegree, GeoTools, and GeoServer⁷ realized the reference implementations for the WMS 1.1.1 and WFS 1.0 specifications.

References

- Cuthbert, Adrian (1998): "User Interaction with Geospatial Data". OpenGIS® Project Document 98-060. (available for OGC members)
- Müller, Markus et al.(2003): "Development of an environmental data infrastructure for Hamburg". In: Albrecht Gnauck, Ralph Heinrich (Eds.): *The Information Society and Enlargement of the European Union*, 17th International Conference Informatik for Environmental Protection, Cottbus 2003. Marburg: Metropolis-Verlag.
- Nebert, Douglas (Ed.) (2001): *Developing Spatial Data Infrastructures: The SDI Cookbook*. Version 1.1 – 15 May 2001.
<http://www.gsdi.org/pubs/cookbook/cookbook0515.pdf>
- [OGC] Open GIS Consortium (1999): *The OpenGIS™ Abstract Specification Topic 5: Features*. Version 4. OpenGIS® Project Document 99-105r2.
<http://www.opengis.org/specs/?page=abstract>
- [OGC] Open GIS Consortium (2003): Request For Quotation And Call For Participation In the OGC Web Services 2 Initiative, Initial Operating Capability and Demonstration, Annex B: OWS-2 Architecture. <http://www.opengis.org/initiatives/?iid=7>

⁶Compliance & Interoperability Testing & Evaluation Initiative. The corresponding portal can be accessed on <http://cite.occamlab.com/>

⁷[http://\[deegree|geoserver|geotools\].sourceforge.net](http://[deegree|geoserver|geotools].sourceforge.net)