

## Research Article

# Using the SDI<sup>light</sup> Approach in Teaching a Geoinformatics Master

**Barend Köbben**

*Faculty of Geo-Information Science  
and Earth Observation  
ITC – University of Twente*

**Rolf de By**

*Faculty of Geo-Information Science  
and Earth Observation  
ITC – University of Twente*

**Theodor Foerster**

*Faculty of Geo-Information Science  
and Earth Observation  
ITC – University of Twente*

**Otto Huisman**

*Faculty of Geo-Information Science  
and Earth Observation  
ITC – University of Twente*

**Rob Lemmens**

*Faculty of Geo-Information Science  
and Earth Observation  
ITC – University of Twente*

**Javier Morales**

*Faculty of Geo-Information Science  
and Earth Observation  
ITC – University of Twente*

### Abstract

ITC, the Faculty of Geo-Information Science and Earth Observation of the University of Twente, is an institute that aims at capacity building and institutional development, specifically in developing countries. In our Geoinformatics curriculum, we emphasise two principles. The first addresses the systematics of purposeful spatial data production and uptake into computerised systems; the second addresses the methodical construction of these computerised systems, applying principles of model – driven architecture, formal specification and transformational design of SDI nodes. The term Spatial Data Infrastructure (SDI) usually denotes large, complex systems, but its principles can also be applied in simple and cost-effective ways. This approach we have called SDI<sup>light</sup> and it is of particular interest to our students that come from developing countries. We work with and build a software stack consisting of free and open source components. To achieve interoperability, we emphasise the use of open standards from the Open Geospatial Consortium and others. In this paper, we explain how our students apply the SDI<sup>light</sup> approach in the Geoinformatics

**Address for correspondence:** Barend Köbben, ITC – University of Twente, Faculty of Geo-Information Science and Earth Observation, PO Box 6, 7500AA Enschede, The Netherlands. E-mail: kobben@itc.nl

Master degree course. An important part of that course is a Case Study Application Building & Programming, in which students apply their knowledge in a ‘real-world’ project, with a focus on geo-information engineering skills. We conclude with a section that evaluates the effectiveness of using the SDI<sup>light</sup> concept in teaching our Geoinformatics Master, and on the more general applicability of the methodology.

## 1 Introduction

ITC, the Faculty of Geo-Information Science and Earth Observation of the University of Twente, is an institute that aims at capacity building and institutional development, specifically in developing countries. ITC is first and foremost an international *educational* institute, based in Enschede (the Netherlands), with almost 60 years of experience in the field of training in a wide range of disciplines, all closely related to Geoinformatics. ITC offers degree courses in various levels (M.Sc., Master and Diploma degrees) as well as various shorter courses and a graduate programme for students pursuing a Ph.D. degree. The courses aim to provide an in-depth study of a particular set of problems relevant to developing countries and emerging economies.

Our students are not the typical university stereotype, but instead are usually mid-career professionals, with an average age of around 34 and a solid background, coupled with several years of working experience, in their chosen field. Consequently, professional relevancy and problem orientation are important factors in the courses, with the emphasis on the application of concepts and skills. The fields in which these concepts and skills are applied vary widely but always include the use of earth observation, collection and management of spatial information, and the development of data integration methods. The subjects of the programmes form a broad spectrum, from Applied Earth Sciences, Governance and Spatial Information Management, Land Administration, Natural Resources Management, Urban Planning and Management, Water Resources and Environmental Management, to Geoinformatics.

## 2 Teaching Geoinformatics at ITC

The courses offered within the Geoinformatics programme differ from those in other programmes. The Geoinformatics courses concentrate more on the *thorough understanding and methodical construction* of GIS and Earth Observation technology, whereas those in other programmes deal foremost with the *application* of that technology in the various fields. The teaching and research methods are therefore founded in engineering science, as opposed to those from the empirical sciences found predominantly in the other programmes.

In the curriculum, we emphasise two principles. The first addresses the systematics of purposeful spatial data production and uptake into computerised systems, and can be found in various modules in the Geoinformatics Master course curriculum (see, e.g. Section 2.3). The second addresses the methodical construction of these computerised systems, applying principles of model – driven architecture, formal specification and transformational design. All in all, the curriculum aims to equip the student with a professional toolset to design and realise robust spatial information systems, and to populate them with appropriate data and metadata.

## 2.1 Methodical Construction of SDI Systems

Under this heading, the student is acquainted with a body of knowledge about and a number of skills in applying methods to construct components of a Spatial Data Infrastructure (SDI).

The terminology that we apply in this context is maintained throughout the curriculum. For instance, we define an ‘SDI’ as a network of collaborating ‘SDI nodes’ that offer services to each other and possibly also to external users, and ‘SDI node’ as a single system node in an SDI network, which archetypically has data, catalog and portrayal services, amongst others (the term SDI, and its relation to our SDI<sup>light</sup> approach, is discussed in some more detail in Section 2.2).

By the components of an SDI, we mean computer systems as tools in providing services that base themselves to some extent on geospatial data. We look at these systems as software stacks that have been purposefully ‘stacked’. Within our SDI<sup>light</sup> approach, we adopted a reference system stack using open source components, and any system realisation can be seen as a conscientious pick from that reference stack.

Emphasis is on building the professional skills to design and realise, and eventually also maintain SDI components, based upon sound theoretical foundations. Obviously, nothing is as practical as a good theory, but we also feel that inclusion of theoretic notions provides stability to the curriculum, and protects it against being swayed by the issues of the day in software capabilities. We are explicitly trying to address the student desire to believe that knowing the appropriate ‘button sequence’ is all that is needed to be a good system engineer.

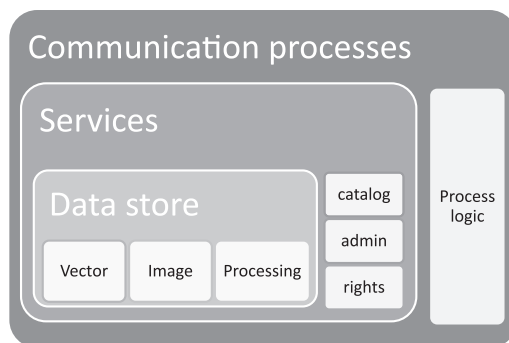
### 2.1.1 General design philosophy taught

Since an SDI is a collaborative network, it follows that the shape of that collaboration needs to be designed and realised. This brings to light the need to understand the *communication processes* between SDI nodes and users. The ‘mouths and ears’ of these processes are various *service offerings and invocations*, as we firmly believe that service-oriented architectures are here to stay within the GI field for a long time to come. The content side to the services, whether produced or consumed, is in the *data store* held by the respective SDI nodes. Hence, the data system also deserves specific design attention.

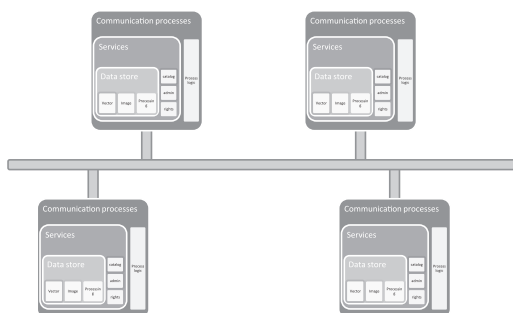
### 2.1.2 Pragmatic design methods

The above description is requirements-centric; it does not necessarily reflect the chronology of design effort. Moreover, a complete SDI is not designed in such a top-down fashion, but it evolves and matures over time. This does not invalidate the above perspective however, and seeing the realisation of SDI communication processes as a goal remains useful. Any designer, and any student in our case, must understand the overall goal definition, but subsequent design work is carried out in a bottom-up fashion. This means that our students approach SDI node design by the following steps:

1. Data store structure
2. Data store functions
3. Catalog and other middleware



**Figure 1** The internals of a single SDI node stack, displaying the various design/realisation points of attention



**Figure 2** An SDI as a network of communicating SDI nodes

4. Service functionality
5. (Communication) Process functionality

Current state-of-the-art in SDI is mature, we feel, in the first three (see, e.g. Kresse and Fadaie 2004, Parent et al. 2006, Rigaux et al. 2002, Shekhar and Chawla 2003) and is developing well into the area of service functionality, although improvements are still required in the transition from syntactic to semantic interoperability of SDI nodes (see, e.g. Craglia et al. 2008, Friis-Christensen et al. 2007). Process functionality has hardly been addressed in scientific papers, but recently authors from our group have published a discussion on this topic (de By et al. 2009). By process functionality, we specifically mean the protocols of communication that must be defined to confine SDI nodes to only sensible interactions between them.

The various steps show up in the simplified stack sketch of Figure 1. Expanding on that view, Figure 2 illustrates an SDI as a network of communicating SDI nodes.

### 2.1.3 Separation of concerns

Complex, data-intensive systems such as most SDI nodes must be methodically designed in phases, where each phase addresses only a single concern. This is a robust and accepted design philosophy, which expresses itself in data-intensive systems based upon the following three concerns:

- **What is the required information content?** In this domain, we look at object populations as classes, describe their characteristics, and how objects can associate. We also describe business rules as constraints over the object model.
- **What is the most appropriate data structure?** Here, we derive a data store model for the object model that sets out a structure to meet the above information requirements. This level also addresses, in part, the options of implementing the indicated constraints with appropriate means.
- **What is optimal performance of the system?** Finally, we address issues of how to optimally realise the data store model. All left-over constraints from the concern above are covered in this phase.

We devote more teaching time on the data side of things than on the function side of things. This is in part a historic bias of the field, but it is also caused by a genuine lack of formalisms that allow us to express functionality at the level of information content, and indeed a little bit at the data structure level. Thorough teachings in SQL, procedural programming, as well as embedded SQL in the curriculum aim to address this issue.

### 2.1.4 Transformational design

The reader may have recognised the levels of conceptual, logical and physical design in the above description. Our students are taught to strictly separate these issues, and express each concern as a separate design document. They are also taught, using formally correct design transformations, how the object model can be transformed into a data store model, and subsequently, how a data store model can be turned into a realised system with stated performance characteristics. With all this comes a body of terminology that they should adhere to.

We use Enterprise Architect, a licensed CASE/UML tool (from Sparx Systems, see <http://www.sparxsystems.com/products/ea/>) to help out in structuring the design/realisation process. This is because there is no open source alternative of comparable functionality. EA is currently spatially agnostic, therefore we are developing support for ISO and OGC spatial data standards as loadable add-ons for it (some of this work is based on van Bennekom-Minnema 2008), thereby allowing students to create object and data store models that rest on ISO/OGC data types, for instance.

## 2.2 The SDI<sup>light</sup> Approach

As mentioned in Section 2.1, the methodical construction methods described above are practised and taught using a reference SDI system stack using open source components, which we call SDI<sup>light</sup>.

The term SDI for Spatial Data Infrastructure (sometimes also GDI for Geo-Data Infrastructure) may be usually connected with (very) large regional or national spatial data warehouses, but more generally is (in the frequently used definition from Groot and McLaughlin 2000) “the networked geospatial databases and data handling facilities, the complex of institutional, organisational, technological, human and economic resources (. . .) facilitating the sharing, access to, and responsible use of geospatial data at an affordable cost for a specific application domain or enterprise.” In many cases SDI data and application infrastructures are being developed using high-end geospatial software

solutions and large corporate databases, requiring substantial investments in financial and human resources. But the principles of SDI can also be applied in more simple and cost-effective ways. This down-to-earth approach was introduced in Köbben (2007), and later worked out further in Köbben et al. (2009).

This approach provides researchers and students alike with a proof-of concept platform for relatively simple, low-cost, yet powerful ways of sharing data amongst various distributed offices and institutions as well as the general public. To achieve these, we use open standards whenever available, open source solutions where possible and commercial software where necessary. Given that a considerable part of our courses are focussed upon training geo-informatics *engineers*, students are involved in actively developing and building (parts of) such systems.

In general, the main building blocks we use are:

- A *spatial database back-end* that stores the spatial data using the Open Geospatial Consortium Simple Features specifications. As a platform, the PostGIS extension (see <http://postgis.refractory.net/>) to the object – relational DBMS PostgreSQL is a logical choice. First, because PostgreSQL is a solid DBMS that has a reasonably gentle learning curve, yet is wonderfully appropriate for advanced database applications, and its documentation is very transparent (PostgreSQL Global Development Group 2008). PostGIS in addition, is the leading open standards implementation of spatial vector management, and enjoys a lively and supportive user/developer community. The combination of the two is enjoying an ever-growing user base, particularly in the domain of industrial applications.
- A set of *interoperable middleware web applications* that interface with the database back-end and with each other, and fulfil tasks such as delivering maps for visualisation purposes or providing data and processing services. We use existing open source solutions, mainly MapServer and GeoServer, and we develop our own components (see Section 2.2.4)
- Simple (thin) *browser-based clients* enabling access to the maps and data. At present, we employ various techniques such as dynamic HTML, asynchronous JavaScript and XML (AJAX) – using the OpenLayers API – and Scalable Vector Graphics (SVG).
- Thick clients: We use QuantumGIS for map viewing and editing, as well as ILWIS open source GIS software.

It is not the intention that SDI<sup>light</sup> should finally become one coherent system, rather it should be seen as a testbed in the broad sense of *equipment for testing*. It is the place where we can show fellow researchers, consultants and students as well as possible users (such as GIS and Cartography departments in developing countries) that the things we teach can be made to work quite quickly, in a relatively simple and low-cost setup. This approach has been functioning well for some five years now.

### 2.2.1 SDI<sup>light</sup> in teaching

The majority of ITC students come from developing countries, and they have a keen interest in open source software. Certainly in the geo-webservices field, the available solutions can compete with proprietary software and in many cases fit our students' needs best. There are of course also parts of the curriculum where proprietary software is better suited, or simply the only viable option, which for example is the case in digital

photogrammetry. Also, one must not forget the valid desire of students (and their employers) to leave ITC with a solid working knowledge of leading GI software, and therefore ‘marketable’ skills. The practical parts of the Core Modules of all ITC courses (see Section 2.3) are executed using ESRI ArcGIS and ERDAS Imagine. There is a site-wide license of these softwares and students receive a complimentary copy of ArcView (including extensions such as Spatial Analyst and 3D Analyst), upon finishing their studies. To make sure the two worlds can be integrated as much as possible, we emphasise the use of OGC and other open standards in all our teaching and practical exercises.

Courses at all levels of the Geoinformatics programme now include a module in which students learn how to set up middleware that serves WMS and WFS. They use those services in existing clients (e.g. QuantumGIS and ESRI’s ArcMap), but also develop their own web-clients using the OpenLayers API. The major use of the SDI<sup>light</sup> stack is in the Case Study Application Building and Programming, that is part of the Master degree, and is described in more detail in Section 2.3.1.

### 2.2.2 SDI<sup>light</sup> in research

The software stack used in SDI<sup>light</sup> is also part of our research work, where we use it both as a tool for prototyping and as a subject of research itself.

An example of the former is TimeMapper, a proof-of-concept for online dissemination of spatiotemporal information. The intention of the TimeMapper system is to generate directly from a spatial database, “on the fly”, interactive and animated vector maps, using the Scalable Vector Graphics standard and its SMIL declarative animation language. A first prototype (as illustrated in Figure 3) was developed for the exploration of moving object dynamics of Antarctic icebergs.

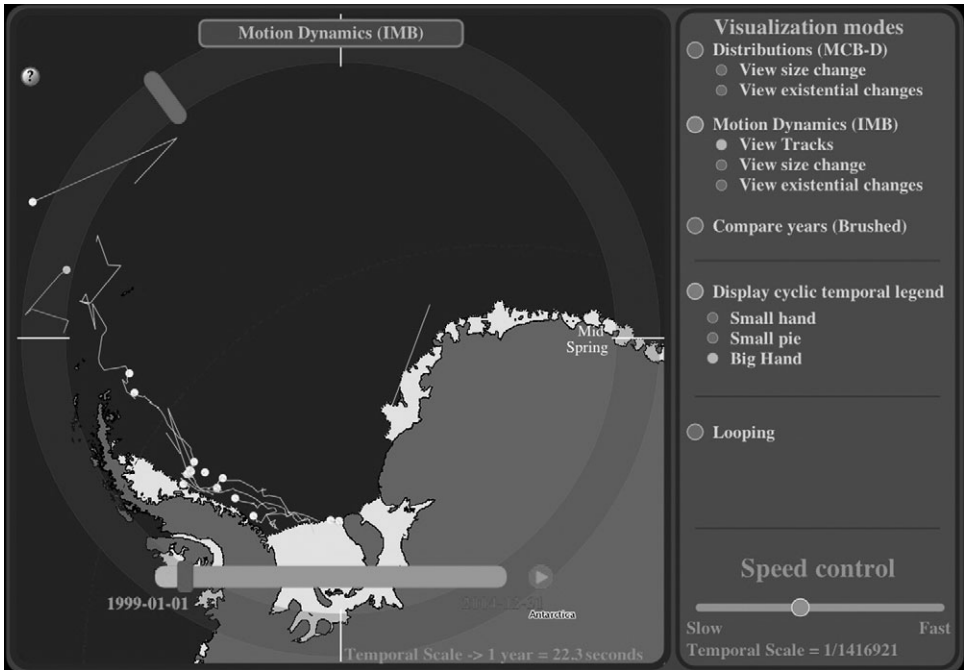
An example of the stack as subject of research is a body of M.Sc. theses written on SDI software stack design using web frameworks like Web2Py and GeoDjango, and on the extension of PostGIS with support for TIN and raster. That last example also shows the flexibility of working within a non-proprietary software environment, as it did not result in a PostGIS solution, because INPE’s TerraLib (see Câmara et al. 2009) was found to be more useful.

### 2.2.3 SDI<sup>light</sup> in projects

One of the benefits of the SDI<sup>light</sup> approach is its direct applicability during the execution of project services. Clients are keen to see an implementation of the ideas that we bring to the table to address their needs. The open source software stack allows for rapid prototyping, so that implementations can be developed at very early stages of a project. This enables clients to better define their interests, and also allows us to better tailor our solutions to their specific context.

Various examples of this scenario exist. One was a EU-funded project that established a *virtual museum* for the archaeological site of Melka Kunture in Ethiopia. For this project we used PostGIS to store the data of close to 50,000 archeological artefacts and make them accessible within maps and information panels of a virtual museum on the Web (<http://geoserver.itc.nl/melkakunture>).

We have also recently been involved in a large project in Mongolia, called “The National Geoinformation Centre for Natural Resource Management.” The overall



**Figure 3** Prototype of interactive animation of 19 icebergs in the Weddel Sea (Antarctica) from January 1, 1999 to December 31, 2004. Interactive original can be loaded from <http://geoserver.itc.nl/TimeMapper/>.

objective of this project is to support environmental policy and decision-making processes, and to contribute to the sustainable development of Mongolia. Here, the full range of the SDI<sup>light</sup> approach can be used: Firstly, the strengthening of the capacity of the agencies responsible by deploying the training modules developed for the Geoinformatics curriculum. Secondly, together with the agencies, implementing basic geo-webservices for their data. Finally, developing tailor-made applications for Fire Management, Drought Management and Management of *Zhud*, a specific Mongolian word for a collection of natural hazards (see the NGIC project website at <http://geodata.mnengic.mn>).

#### 2.2.4 SDI<sup>light</sup> component development

Besides the development work in various research and project efforts, we also participate actively in a coordinated development of software components, in the framework of 52°North, an international research and development initiative, whose mission is to promote the conception, development and application of free open source geo-software for research, education, training and practical use (see <http://www.52north.org/>). Here we are especially active in the development of ILWIS, a complete package for image processing, spatial analysis and digital mapping. In the 22 years since this software was first released by ITC, it has established a wide user community (see its website at <http://www.52north.org/ilwis>). Furthermore we are involved in 52°North's Geoprocessing and Sensor Web Communities.

Weeks	Modules	In parallel
1-12: Core Modules	1: Principles of Databases	Programming Skills
	2: Principles of Remote Sensing	
	3: Principles of Geographic Information Systems	
	4: Mathematics and Programming	
13-30: Master Modules	5: Sensor orientation & 3D data acquisition	Case Study Application Building & Programming
	6: Spatial Data Modelling	
	7: Base mapping from images	
	8: SDI Engineering	
	9: Process modelling & spatial analysis	
	10: Dissemination & visualization of geospatial data	
31-34	11: Advanced Topics and Master Skills	
35-37	12: IFA preparation, Excursion	
38-47	Individual Final Assignment	

**Figure 4** The general setup of the Geoinformatics Master at ITC

### 2.3 Setup of the Geoinformatics Master

The teaching principles and the use of the SDI<sup>light</sup> approach, described in the previous sections, are relevant to any of the degree and other courses within the Geoinformatics programme. In this section we focus on how the principles are applied in the factual setup of the Geoinformatics Professional Master course. Its general setup can be seen in Figure 4.

ITC programmes are modular in structure. A module deals with one subject or related subjects and usually lasts for three weeks. Modules are grouped together in blocks that form a coherent part of the course, and they are:

**Modules 1–4: Core Modules** that are dealing with the principles of geo-information science and earth observation, and taught largely the same in all ITC programmes. We are using a set of ITC-authored textbooks (Huisman and de By 2009, Tempfli et al. 2009), so that students share the same core of GIScience knowledge.

The remainder of the modules are specific to the Geoinformatics Master. In this degree course, we aim to train students in the use of GIScience to generate, analyse and disseminate spatial data, and to be able to apply both conceptual and operational knowledge to manage that process in a professional practice. In this it differs from the Master of Science course that has a stronger emphasis on the scientific discipline, and less on the practical skills. The Master Modules basically follow the pattern of geoinformation production, from data gathering using remote sensing, through constructing data models, populating these models with gathered and processed data, to disseminating the information to the users:

**Module 5: Sensor orientation & 3D data acquisition** deals with data acquisition based on sensors: the positioning and orientation systems, the types of and qualities of different cameras, laser scanners, and SRTM and the geometric processing of raw sensor data to obtain primary 3D data. The result of these processes are products that can serve as a spatial data component in an SDI, and as a basis for object recognition and reconstruction procedures.

**Module 6: Spatial Data Modelling** focusses on spatial database design and the application of spatial database technology. This is firmly based on the use of appropriate standards and the principles of model-driven architecture, formal specification and transformational design, as highlighted in Section 2.1. The spatial data components produced in module 5 have at the end of this module found a home in well-designed spatial databases.

**Module 7: Base mapping from images** takes the data produced in modules 5 and 6 further to object recognition and reconstruction procedures. The focus is on acquiring and updating vector data from imagery, based on the work flow of first analysing data requirements, then extracting features from optical sensor data, and finally validating the data produced. At the end, the well-designed spatial databases of module 6 have validated, up-to-date contents.

**Module 8: SDI Engineering** introduces the core technologies of SDI systems: interoperability, client–server architecture, XML, geo–websites and open standards (specifically those of the Open Geospatial Consortium). It also familiarises the students with the SDI<sup>light</sup> approach we introduced in Section 2.2, giving them a design (over)view as well as practical working skills with the various software packages involved. The outcome of this module is that the spatial data of the earlier modules is now available in a spatial data webservice infrastructure, adhering to the SDI<sup>light</sup> principles.

**Module 9: Process modelling & spatial analysis** deals with the design and implementation of geo–processing models. These models can contain a series of analytical operations, can be re–executed or automated and should be stored and shared in a well–described infrastructure. A main goal is for students to obtain a good understanding of the inherent uncertainties of the process models. The outcome of the module is that the spatial data webservice infrastructure is enhanced with geo–processing capabilities, and the results of this processing are available within the same infrastructure.

**Module 10: Dissemination & visualisation of geospatial data** finally introduces the students to dissemination methods and environments. Students learn about user-centered design and usability research as important starting points for spatial data dissemination. Attention is also focussed upon cartographic design principles and the practical application of the theory of geovisualisation. Within the SDI<sup>light</sup> stack, the students use the knowledge and skills gathered in this module, to build usable web–based interfaces for their spatial data infrastructure.

Although in the modules as described above, the students are exposed to all steps of geoinformation production, it is inevitable that this process is somewhat fragmented and rather theoretical in places. We felt there was a need for newly–gained knowledge and skills to be ‘grounded’ by having students build one coherent system from start to finish by themselves. As can be seen in Figure 4, these activities are grouped into in a separate module called *Case Study Application Building & Programming*, conducted in parallel to the Master Modules.

### 2.3.1 *Module on case study application building & programming*

The main objective for this Case Study module is for students to apply their knowledge and skills in a ‘real–world’ project that focusses on geoinformation *engineering* skills.

Within structured periods of time, they are required to build a coherent system for spatial data acquisition, storage, access, analysis and dissemination, that complies as much to SDI qualities and standards as can practically be implemented within the available resources and time.

The use case for this module is a primary school planning system: Students assume the role of a Geoinformatics specialist who is asked by the municipality of Enschede, to develop a planning system that can be used to evaluate the current public primary schools and to plan the location of new schools.

The overall aim is to mimic a typical consulting job, in which the teacher assumes the role of the municipal contractor. A system requirements document is drafted as a set of technical and functional requirements, written by the prospective user of such a system: a municipal school planner that requires access to spatial data about primary schools within the Enschede municipal boundary, and the ability to:

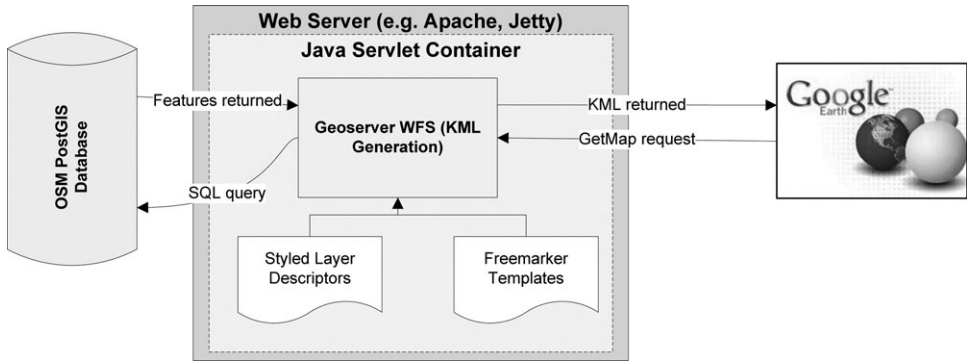
- explore key data of schools and their service area:
  - locations of schools and key data (e.g. number of pupils);
  - surrounding neighbourhood(s) and their number of inhabitants, potential pupils, etc.;
  - connections and distances to school (e.g. by car, bicycle, foot);
  - proximity to other schools, to other objects (e.g. polluting industry).
- study school location suitability:
  - for (re)locating existing or new schools;
  - taking into account policy concerns like accessibility, safety, etc.

The basic technical requirements for the system are also set: it is to be used on the Internet, should use open standards (as set by the W3C and the OGC), wherever possible, and it should be usable for a non-specialist, someone with only basic computer skills and no special GIS training.

The system is developed step by step, and students submit each part separately so that the contractor can provide comments. These comments are then incorporated in the next step of the system development. In real life, these steps are the separate Case Study days in the Geoinformatics Master, usually 2–3 days per module. Students are asked at several moments to hand in intermediate reports, and show and explain functioning parts of the system to the contractor. The emphasis is on the data infrastructure – to provide the planner with simple, platform-independent interfaces to spatial applications. The interface of this application should be designed in a user-centric manner, and provide appropriate visualisations. In order to be successful, the students are required to employ the knowledge and skills learned in all of the modules to actually implement the SDI<sup>light</sup> approach in a working system.

### 3 Conclusions

The Case Study was introduced in 2004, the same year that the Geoinformatics Master was set up in its current form. The reason for the new setup as well as the introduction of the Case Study was to provide the students with a more integrated view of Geoinformatics, based on the SDI paradigm, and the goal was for the course to provide a greater focus on engineering skills. Although we have no hard facts or figures to comment on the effectiveness of that move, several measures indicate we have succeeded in that goal: The



**Figure 5** The architecture of a system to deliver OSM data to Google Earth, from the (unpublished) Master assignment report of M. Graham (discussed in Köbben and Graham 2009)

Master has been attracting a steady input of 12–24 students yearly, in the same period when other Master courses at ITC have declined steadily in numbers, and been abolished. In the final projects completed by students, we have seen an increase in projects dealing with exactly those subjects we discussed in this article: methodical design of SDI nodes, open source technology in spatial databases, web-services and -clients, etc. One example of such a project is the work reflected in Figure 5: This student has delivered crowd-sourced data of Open Street Map to the Google Earth interface, by building intermediate web services out of SDI<sup>light</sup> components.

Other courses have also shown interest in the SDI<sup>light</sup> approach: As mentioned in Section 2.2.1, all levels of the Geoinformatics programme now include a module in which the basic technology is introduced. It also has been included in a Bachelor Minor that is offered to students of all programmes of the University of Twente and in a distance learning part-time M.Sc. conducted in cooperation with the Universities of Utrecht, Wageningen and Delft. Furthermore, we have attracted external funding for successful courses on the subject in cooperation with the University of KwaZulu-Natal (South Africa) and the IGAC/CIAG agency in Bogotá (Columbia).

Whether one deals with small projects with a limited scope, such as the case study use case described above, or with complex national SDIs such as the Mongolia project mentioned in Section 2.2.3, a thorough grounding in GI science concepts and a solid construction method are always required. Regardless of the software solution chosen, a well-understood and proper design methodology has to be employed. That is what we teach our students and what we try to practise in our own research and project work. In addition to a thorough theoretical understanding, one needs an appropriate set of tools, and experience with using those tools in various settings. That we have found in our SDI<sup>light</sup> approach. In recent years, it has become an integral part of geo-informatics teaching, research and projects at ITC.

## References

- van Bennekom-Minnema J 2008 The Land Administration Domain Model 'Survey Package' and Model Driven Architecture. Unpublished MSc Thesis, Joint Programme in Geographical Information Management and Applications (GIMA), Enschede/Utrecht/Wageningen/Delft

- de By R A, Lemmens R, and Morales J 2009 A skeleton design theory for spatial data infrastructure. *Earth Science Informatics* 2: 299–313
- Câmara G, Vinhas L, Ferreira K R, de Queiroz G B, de Souza R C M, Monteiro A M, de Carvalho M T, Casanova M A, and de Freitas U B 2009 TerraLib: An open source GIS library for large-scale environmental and socio-economic applications. In Hall G B and Leahy M G (eds) *Open Source Approaches to Spatial Data Handling*. Berlin, Springer-Verlag: 247–70
- Craglia M, Goodchild M F, Annoni A, Camara G, Gould M, Kuhn W, Mark D, Masser I, Maguire D, Liang S, and Parsons E 2008 Next-generation digital earth. *International Journal of Spatial Data Infrastructures Research* 3: 146–67
- Friis-Christensen A, Lutz M, Ostländer N, and Bernard L 2007 Designing service architectures for distributed geoprocessing: Challenges and future directions. *Transactions in GIS* 11: 799–818
- Groot R and McLaughlin J 2000 *Geospatial Data Infrastructure—Concepts, Cases and Good Practice*. Oxford, Oxford University Press
- Huisman O and de By R A (eds) 2009 *Principles of Geographic Information Systems: An Introductory Textbook* (Fourth Edition). Enschede, ITC Educational Textbook Series No 1
- Köbben B 2007 RIMapperWMS: A web map service providing SVG maps with a built-in client. In Fabrikant S and Wachowicz M (eds) *The European Information Society: Leading the Way with Geoinformation*. Berlin, Springer-Verlag Lecture Notes in Geoinformation and Cartography: 217–30
- Köbben B and Graham M 2009 Maps and mash-ups: The national atlas and Google Earth in a geo-webservices infrastructure. In *Proceedings of the Twelfth AGILE International Conference on Geographic Information Science*, Hannover, Germany
- Köbben B, Lemmens R, Morales J, de By R A, and Foerster T 2009 The SDI-light OSGEO stack at ITC. In *Proceedings of the Open Source GIS UK Conference*, Nottingham, United Kingdom
- Kresse W and Fadaie K 2004 *ISO Standards for Geographic Information*. Berlin, Springer-Verlag
- Parent C, Spaccapietra S, and Zimányi E 2006 *Conceptual Modeling for Traditional and Spatio-Temporal Applications: The MADS Approach*. Berlin, Springer-Verlag
- PostgreSQL Global Development Group 2008 PostgreSQL 8.3.7 Documentation. WWW document, <http://www.postgresql.org/docs/8.3/static/release-8-3-7.html>
- Rigaux P, Scholl M, and Voisard A 2002 *Spatial Databases with Application to GIS*. San Francisco, CA, Morgan Kaufmann
- Shekhar S and Chawla S 2003 *Spatial Databases: A Tour*. Englewood Cliffs, NJ, Prentice-Hall
- Tempfli K, Huurneman G C, Bakker W H, and Janssen L L F (eds) 2009 *Principles of Remote Sensing: An Introductory Textbook* (Fourth Edition). Enschede, ITC Educational Textbook Series No 2