XML Technologies and Geodata

Peter DORNINGER

(Dipl.-Ing. Peter Dorninger, Vienna University of Technology, Institute of Photogrammetry and Remote Sensing, Gusshausstrasse 27-29, 1040 Wien)

1 INTRODUCTION

Photogrammetry and Remote Sensing are very important methods for acquisition of geodata. During the previous decade, several revolutionary changes occurred in this area. Until the appearance of automated image analysis tools, it was necessary to measure selected points in the images given. At that time, it was much faster and even cheaper to get images of real world objects compared to the time and money consuming process of manual analyses. So one tried to minimize this effort by measuring only characteristic points such as edges, break-lines, peaks and valleys and, for sure, a grid with a given grid step which was selected to meet the efforts. Lots of information in the images was neglected.

Digital *point matching algorithms* and *airborne laser scanning* provide many new possibilities. The only restriction on spatial resolution is the one of the used sensors. Given a more precise image sensor, the matching algorithm will be able to match more surface points; given a higher frequency laser scanner, more points can be measured of the same area. And those sensors get more and more precise every day. Besides, those techniques allow for fast repetition which is necessary to create time series as a basis for 4D modeling!

However, this fact is accompanied by several problems concerning the capability of available computers. Some years ago, as the first ideas of 3D city models arose, it was very difficult to acquire the necessary data. Today the new sensors and methods have the necessary capability, but we are not able to handle the available datasets efficiently, because of shortcomings in the past. In a time of world wide data exchange through the internet and global datasets, it is necessary to have efficient methods and algorithms to manage the available data. There is a need for international, vendor independent data exchange and management standards that have to be accepted and supported by the industry.

This article is going to present several methods of data encoding using standardized data formats based on *eXtensible Markup Language (XML)*. After an introduction to this kind of data encoding, two derived applications for management, storage and presentation of geodata are described. As XML data is written in text format, the datasets have the ability to become rather long. Therefore some promising methods to reduce the amount of data are introduced afterwards. XML documents are mainly used for data exchange between databases. Therefore the capabilities of commonly used database systems for storage of geodata are described in the end and current implementation results of the Institute of Photogrammetry and Remote Sensing (I.P.F.) are presented.

2 XML BASED DATA EXCHANGE FORMATS

During the past few years, a new standard for data exchange has been defined, the *eXtensible Markup Language (XML)*. XML is not one predefined data format such as DXF (Autocad Exchange Format) or VRML (Virtual Reality Modeling Language). It is a structural and semantic language which allows for description the encoded information. As it is a meta language, it allows for defining other markup languages such as Mathematical Markup Language (MathML), Chemical Markup Language (CML) or for example Geography Markup Language (GML), a standardized XML based format for the management of geodata.

Most XML based standards and recommendation are defined open and concern independent. They are worked out by independent commissions and organizations such as the World Wide Web Consortium (W3C – http://www.w3.org) or the OpenGIS Consortium (OGC – http://www.opengis.org/) in which many different companies are participating. This provides that changes in the specification can not be caused on the goodwill of only one company. That fact should guarantee a longer life circle of a format. On the other hand this can be seen as one of the lacks because the participating companies can not be forced to use those standards. They often participate during the standardization process in order to get new ideas and finally start to implement their own quasi standards. In the past such things happened several times. Remembering the differences between JavaScript and JScript or actually the development in the acceptance of Scalable Vector Graphics (SVG – a standardized XML application which will be described in chapter 4.2 in further detail) and the Microsoft implementation of Vector Markup Language (VML), a Microsoft quasi standard, although Microsoft participated in the W3C standardization group for SVG.

2.1 A Short Introduction to XML

XML used as exchange format for the internet tries to overcome the problems of HTML (Hypertext Markup Language). These are limitation (the HTML Document Type Definition (DTD) is fixed and does not allow for any kind of extension), mixture of formatting information and data (XML separates them) and it is especially built for reusability. Another advantage of XML is the support of the Unicode character set, a key issue for internationalization.

Generally speaking, an XML document consists of three different logical structures: *Tags, Attributes* and *Data*. A simple document can look like the following example:

```
<person gender=”male”>
  <firstname>Peter</firstname>
  <lastname>Dorninger</lastname>
</person>
```

Tag names are written between ‘<’ and ‘>’ and either occur pair wise as start and end tag or, for empty elements, as an empty element tag. Start tags may include attributes. The corresponding value has to be limited with quotes. So the data can be stored as attribute values as well. However, normally attributes are used to hold some meta information which is used for several purposes (e.g.: language information) by the parser that interprets the document.

The valid structure of an XML document can be defined using two different mechanisms: *Document Type Definitions (DTD)* and *XML Schema Definitions (XSD)*. As XML Schemas are standardized by the W3C and they are defined in XML and provide more flexibility than DTDs, they are generally used to define XML applications.

There are many other XML related technologies. The following list gives an overview of some of the most important ones. A lot of literature is available for further studies (http://www.w3.org/XML/).

Namespaces	allow to restrict the scope of validity of XML objects
XSD – eXtensible Schema Description Language	defines valid structures and datatypes of an XML document
XQuery	query language for XML documents (interface to databases)
XLink	allows for linking of documents (extended HTML hyperlink functionality)
XPath	object selection language
XSL – eXtensible Stylesheet Language	describes how to display / render documents; it consists of two parts: *XSLT:* defines the transformation from one XML format into another *XSL:FO:* formatting objects – allow to format an XML document

Tab. 1: XML related technologies

The acceptance and application of those standards in the industry is growing. XML used as data exchange format has reached a good position. Although some of the mentioned technologies are brand new, they get more and more important due to the fact that they are already implemented in standard software such as Microsoft Internet Explorer and ready to be used. For sure there is still a large area of research. For example, the current implementations of XQuery have big performance shortcomings. As Gottlob [2002] shows in his tests, the evaluation time grows exponential with the complexity of a query and so they can only be analyzed to a particular length. Some theoretical approaches to solve this problem are described in this paper as well.

3 XML AND GEODATA

Currently several promising XML standards concerning geodata do exist. In the following the main features of two already standardized geodata XML applications are described: *Geography Markup Language (GML)* and *Scalable Vector Graphics (SVG)*.

3.1 Geography Markup Language

In order to define one single data format for inter-vendor geodata exchange, the *OpenGIS Consortium (OGC)* was founded in 1994. The resulting OGC's OpenGIS Interface Specifications have become a successful foundation for interoperable geodata processing (e.g.: OpenGIS Simple Feature Specification). During the following years the importance of the World Wide Web (WWW) as data exchange medium and the development of XML as proprietary data exchange format resulted in the development of *GML*, an OGC recommendation of an XML application which realizes the OGC Interface Specifications using XML. The following definition describes GML in a good manner:

> *Geography Markup Language (GML) is an XML extension for encoding the transport and storage of geographic information, which includes geometry and properties of geographic features [Reichardt 2001].*

In this definition the expression 'XML extension' is a synonym for 'XML application'. An online version of the OGC Implementation Specification of GML can be found at the OGC web page [OGC 2001].

3.1.1 Key Issues of GML

The GML standard enables to model the world according to the OGC Abstract Specifications, which define a geographic feature as "*an abstraction of a real world phenomenon; it is a geographic feature if it is associated with a location relative to the Earth.*" [OGC 2001]. The GML specification is concerned with the OGC Simple Features, features whose geometry properties are restricted to 'simple'. Point, line, linestring or arc are good examples for such objects. Unlike the simple feature specification, GML allows for 3D coordinates, but it does not directly support 3D geometry constructs. By the way GML is fully compliant to the XML Schema and Namespace Recommendation. The general definition of GML is done using a set of three base schemas:

- *Feature Schema* - *feature.xsd:* defines the general feature-property model
- *Geometry Schema* - *geometry.xsd:* includes the detailed geometry components
- *XLink Schema* - *xlink.xsd:* provides the XLink attributes used to implement linking functionality

As one of the key issues of XML applications, GML supports full extensibility. So it allows for definition of individual types by extending the given schemas. Another benefit of XML is the separation of content and presentation. Therefore GML does not regard how to visualize the stored data. But as described in former chapters, there are several XML relevant mechanisms to transform GML documents into comfortable visualization formats such as SVG or X3D, an XML application that can easily be transformed into VRML (Virtual Reality Modeling Language), the commonly used visualization format for web presentation of 3D data.

3.1.2 Advantages and Shortcomings

GML as an XML application provides the main advantages of XML such as vendor independency, interoperability, extensibility and many more. In the current status (Version 2.0) it is realized to support the properties of the OGC Simple Features with some extensions such as the capability to handle 3D coordinates.

The possibilities to manage complex geometry data types is rather limited considering only the given definitions. But as GML supports extensibility, complex data types can be defined using collections of simple ones. Regarding large topographical datasets such as point clouds, problems due to the large overhead within XML datasets do occur, caused by the numerous tags and the included meta information. Some promising solutions to this problems will be described within the chapters 4.3 and 4.4.

3.2 Scalable Vector Graphics

A very efficient XML standard for storage, exchange and especially web presentation of geodata is *Scalable Vector Graphics (SVG)* which is standardized since September 2001 by the W3C. The current version of the specification can be found on the web [W3C 2001].

SVG allows for interactivity and animation and it is compatible to other XML standards such as DOM (Document Object Model – supports access of the objects entities), XSL (XML Stylesheet Language), SMIL (Synchronized Multimedia Integration Language) and many others. So it is no problem to transform a GML document, which can be the result of a database request for example, into a valid SVG document using XSLT in order to display the resulting document in a common web browser.

3.2.1 Geometry Types

As *Vector* is part of the standards name, it has the capability to store vector data. However, there are three different groups of object types that can be represented using this standard:

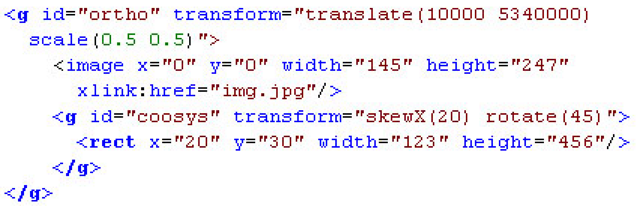
- Vector Data
- Raster Images
- Text

The following basis *vector geometry* types are defined: *rectangle, circle, ellipse, line, polyline, polygon, symbol and path*. Within this set of object types, path is the most complex one, because it has the capability to represent line segments, square and cubic Bezier curves and arc segments. To each object type different style attributes can be defined such as line-style, weight, color and so on. The integration of *raster overlays* allows to use raster images in combination with the corresponding vector data. They can be treated like any other geometry object using the same methods. *Text objects* have the same status as any other basic geometry type. They can be modified using the same attributes.

3.2.2 Special Features

SVG provides numerous methods to visualize and render spatial objects. Besides to line-style and color definitions, sophisticated filter methods are implemented that can be applied on raster overlays as well as on vector elements, due to the fact that these are converted into raster format for visualization purposes. Figure 1 shows an SVG example to give an impression on the capabilities of these methods. Another important feature is the capability of SVG to store georeferencing information and to define coordinate transformations from one spatial reference frame to another. This allows to visualize the stored entities as cartographic products. Together with its definitions for animation and interactivity, it provides all the necessary tools to realize sophisticated web maps. Figure 2 shows the capabilities to georeference and transform SVG objects using the provided methods. Mathematically, all transformations can be represented as 3x3 transformation matrices. Nesting of transformations is allowed.

Fig. 1: Rendering Features of SVG



```
<g id="ortho" transform="translate(10000 5340000)
  scale(0.5 0.5)">
    <image x="0" y="0" width="145" height="247"
      xlink:href="img.jpg"/>
    <g id="coosys" transform="skewX(20) rotate(45)">
      <rect x="20" y="30" width="123" height="456"/>
    </g>
</g>
```

Fig. 2: Georeferencing and transformation of SVG objects

3.2.3 On the fly generation of SVG

There are many different methods that allow for efficient access and manipulation of SVG documents. In the following some Java Technologies that can be used for on the fly (OTF) generation of such documents are described.

JavaServerPages (JSP) allow to create dynamic HTML pages. Conventional HTML pages are static. Some rather restricted methods to apply interactivity and dynamic processes to a web page are supported by scripting language like JavaScript which is a sub set of core Java. For example, it can be used to define simple functions, to evaluate input parameters or for visualization purposes. JSP allow for including pure Java code with all its capabilities into a conventional HTML source. To compile and interpret this functionality, a so-called JSP and Servlet container is necessary. For example, the Tompcat Servlet Container, which is part of Apaches Jakarta Project, is an open source Java implementation of such a tool (http://jakarta.apache.org/tomcat/). *Servlets* are Java classes which can be called from a simple HTTP request. The given input parameters can be processed and the result is provided as pure HTML code which is replied to the client and can be displayed using browser functionality. In the same way, dynamically created SVG documents can be generated. Every JavaServerPage is translated into a Servlet before it is performed. This requires a few seconds at the first request. From then on a request is directly sent to the corresponding Servlet class. Therefore full Java functionality is supported as a Servlet is able to call external methods of other classes. It allows for reuse of already implemented methods that have been used and tested within other applications so far.

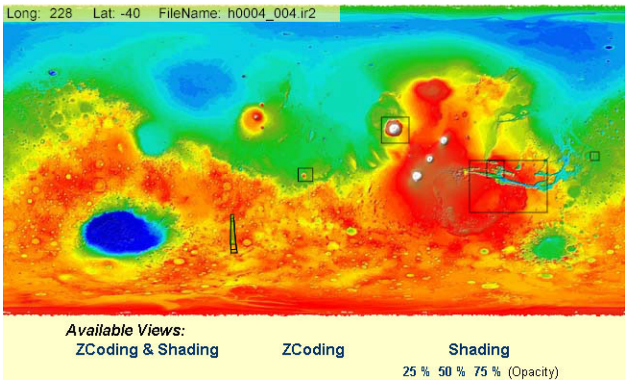
Java Database Connectivity (JDBC) is an API (Application Programming Interface) that allows for access of tabular data sources such as relational databases. It is based on a driver management.

Those two technologies allow for OTF generation of Scalable Vector Graphics based on relational data repositories. Such an SVG is able to handle interactive triggering of DB requests.

3.2.4 SVG Examples

Topographic Mars Information System (TMIS):

Currently the I.P.F. is involved as Co-Investigator into the ESA (European Space Agency) project *Mars Express* which will send a satellite to our red neighbor planet in 2003 (http://sci.esa.int/marsexpress/). The financing is done by the Federal Ministry for Traffic, Innovation and Technology (GZ 190.174/2-V/B/10/2000). The main intention of the working group HRSC (High Resolution Stereo Camera) on Mars Express is to enable a high detailed stereo visualization in color of the whole planet. The I.P.F. has the task to implement an Information System named TMIS which enables to manage and distribute the large amount of image and derived surface point data as well as the computed digital terrain models (DTM). The graphical web interface will be realized using SVG. Figure 3 shows a prototype implementation of the TMIS GUI. The used raster overlay is a visualization of a DTM of the whole Mars surface. Detailed information can be found at the institutes project web page (http://www.ipf.tuwien.ac.at/MarsExpress/).

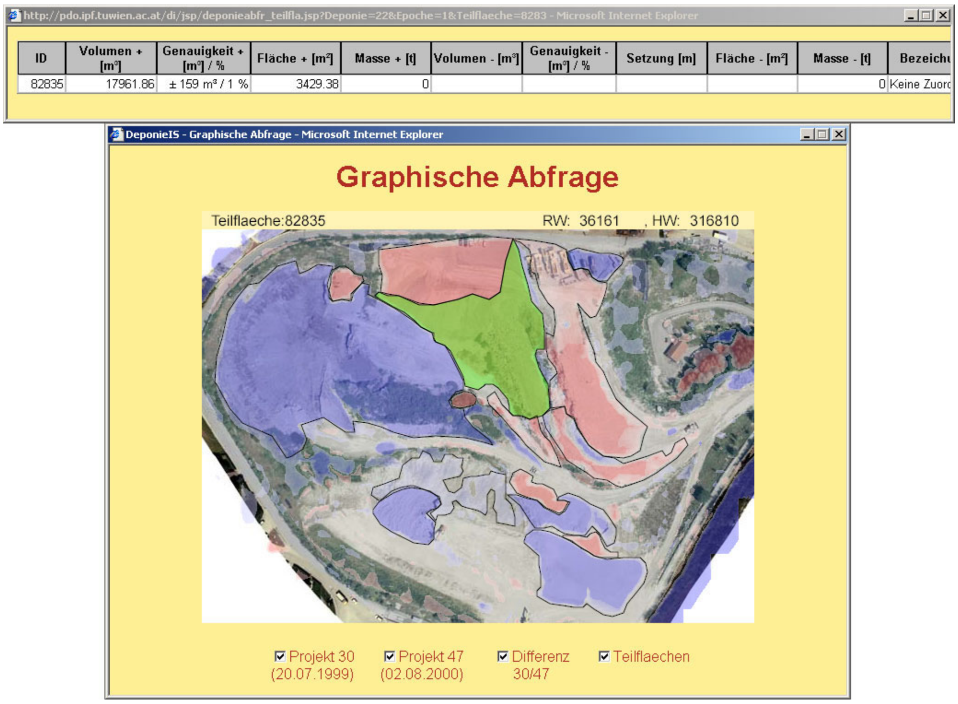


Long: 228 Lat: -40 FileName: h0004_004.ir2

Fig. 3: Prototype GUI of the Topographic Mars Information System (TMIS)

Waste Disposal Information System:

The implementation of the Web GUI of a waste disposal information system [Kraus 2000], another project of the I.P.F., is realized using SVG as well. Figure 4 shows the available data of a waste disposal site. This consists of the orthoimage, a color coding of increasing and decreasing areas and path representations of boundaries of areas of interest. Moving the mouse over such an area provides a short description of it, and clicking into it requests more detailed information from the database. The corresponding SVG is created on the fly using JavaServerPages. The data repository is realized using a Microsoft Access Database.



http://pdo.ipf.tuwien.ac.at/di/jsp/deponieabfr_teilfla.jsp?Deponie=22&Epoche=1&Teilflaeche=8283 - Microsoft Internet Explorer

ID	Volumen + [m³]	Genauigkeit + [m³] / %	Fläche + [m²]	Masse + [t]	Volumen - [m³]	Genauigkeit - [m³] / %	Setzung [m]	Fläche - [m²]	Masse - [t]	Bezeichu
82835	17961.86	± 159 m³ / 1 %	3429.38	0					0	Keine Zuor

DeponieIS - Graphische Abfrage - Microsoft Internet Explorer

Graphische Abfrage

Teilflaeche:82835 RW: 36161 , HW: 316810

☑ Projekt 30 (20.07.1999)
☑ Projekt 47 (02.08.2000)
☑ Differenz 30/47
☑ Teilflaechen

Fig. 4: GUI of the Waste Disposal Information System; here a graphical inquiry is shown

Tirol Atlas:
Another promising long term project by the Department of Geography, Innsbruck University, which is based on SVG is the Interactive 'Tirol Atlas', a transnational Interreg III A project, co-financed by the Land Tyrol and the Autonomous Province of Bozen - South Tyrol. The project period is from 2001 to 2007. Currently there is a first web prototype available (http://tirolatlas.uibk.ac.at/), which will be extended and actualized in content as well as in technical features during the next years. The content is rather exemplary and cannot claim completeness yet. A presentation was given at the SVGOpen in Zurich, 2002. The corresponding paper can be found at: http://www.svgopen.org/papers/2002/foerster_winter__atlas_of_tyrol/

3.2.5 Advantages and Shortcomings

Scalable Vector Graphics have many abilities to become the standard for geodata representation and visualization on the web. It provides many useful features to handle this kind of data in a quite comfortable way. It is an open and vendor independent standard, and software implementations to enable common used browsers to display such datasets do exist. But they are only available as plug-in provided by Adobe, one of the SVG standardization group members. As long as the SVG functionality is not included into the browsers kernel, it is not available to everyone using this software – considering the problems of installing a plug-in on a computer where the current user has no administrators right – and it is rather slow; a real bad ability considering larger datasets.

Within the community of web cartographers there is a large acceptance for this standard. Especially the Swiss Federal Institute of Technology in Zurich, Switzerland, forces its development and application. They established a yearly conference, the SVGOpen (http://www.svgopen.org), which had about 150 participants in the year 2002. But as long as there are no outstanding software solutions that force the developers of web browsers to include SVG viewers into their standard systems, it will not be done. And as long as this viewing functionality is not available to everyone, most software developers hesitate to invest lots of implementation efforts because they are not sure, if the given standard will be accepted. This is a vicious circle which might have the capability to ruin the impressive capabilities of a standard like SVG. Therefore further work and development has do be done to bring this standard to the public.

Another shortcoming is the fact that SVG only supports 2D geometry elements. For cartographic visualizations in the web, this does not matter. But SVG is no usable format for management and storage of 3D geometries.

3.3 Compressing XML

XML is stored in readable ASCII-Format and every dataset is described using tags. Therefore a large overhead occurs within the datasets. Considering several Gigabytes of data which might occur within one single laser scanning project, this fact will cause problems. For example, 600 Millions of points stored in latitude / longitude / height at a precision of 5 digits in lat / long and 3 in height require 16,8 GB disk space without XML tags; converting this point list into an appropriate XML document might increase the amount of data by a factor of two, at least[1].

The transformation and interpolation of the irregular original points into a regular grid array that can be stored efficiently as binary data would reduce this problem. But the high detailed information that is available within the original points is lost. So in many cases this is not a sufficient solution. Nevertheless some considerations on this topic are described in the following chapter.

Therefore data compressing might be a better choice to solve this problem. But as data compression is a rather time consuming process, this will only be a good deal for non time critical problems. There are several well known and often implemented compression algorithms such as *Huffman Encoding* or *Lempel-Ziv Compression* (often mentioned as LZ1 or LZ2 compression which are shortcuts of the two widely used implementations of this algorithm). Both algorithms are lossless, for sure, as lossy compression only makes sense in the case of image, video or audio data compressions for certain applications.

Compression algorithms, as the two mentioned above, do a good job on ASCII data and so they also have the capability to compress an XML dataset in an acceptable manner. But as an XML document has a very large overhead, the compressed data set is still larger than the compressed original text file by an amount of about 20 %. So some further pre-processing is necessary.

As one can imagine, there are many similar structures within an XML document. So it is a good idea to sort the elements of the document according to their tag name. This process is called *reversible transformation*. As we do not want any kind of loss of data, this has to be reversible for sure. Afterwards several other steps are applied. For example, the tag names are stored within a table only once and instead of the original tag name only a unique identifier replaces them within the document. After applying such pre-processing methods, a compression algorithm is applied to this transformed data.

One might argue, that this pre-processing is a very memory and CPU capacity intensive process and that is true. To solve this problem, good implementations of XML compression tools allow for splitting a large document into smaller parts and process them sequentially. Further detailed descriptions of those algorithms as well as test results are given by Mertz [Mertz 2001].

An example of an efficient compressor for XML is the software product XMill. This free and open source software is available on the web (http://www.research.att.com/sw/tools/xmill/). The amazing fact on this implementation is, that the compressed result of an XML document might have only half of the size of the compressed original text file without XML tags! This is enabled, because the used compression method relies on runlength encoding as well. This kind of algorithm is able to compress equal entries if they follow each other. As the XML document was resorted according to its tag names, it is very likely that equal entries are grouped together.

---

[1] The laser profiler MOLA (Mars Orbital Laser Altimeter) was part of NASA's MGS (Mars Global Surveyor) mission (1997-2001). The resulting point cloud of the whole Mars surface (corresponds to the continental area of the Earth) consists of about this amount of points. It allows for the computation of a surface model with 500 m grid width.

3.4 Binary XML

Another possibility to get rid of the overhead of metadata within XML documents is to combine the advantages of XML data description and binary data storage. Such XML applications are called *Binary XML*. Most common data formats such as TIFF or MPEG store some descriptive information in a binary header which is mostly stored as prefix of the actual dataset. This meta information describes for example the extension, the color table, the source or many other data relevant things.

Binary XML uses an XML document to store this descriptive information in ASCII format and the main dataset is either included as binary data stream within this XML document or it is referenced using a link within a tag. External storage of the binary dataset provides the possibility of direct data access using efficient data manipulation routines.

Unfortunately there are no specifications on Binary XML neither in the GML nor in the SVG specification for storage of geometry features in binary format. Obviously raster images are binary data repositories that can be linked to SVG documents using XLink. Nevertheless, it would be a good opportunity to extend these standards for binary geodata handling and enable efficient use of XML dealing with large topographic datasets that mostly consist of gridded point data. In that case, the rather short meta data information could be stored using standardized XML formats.

4 DATABASES AND GEODATA

The previous chapters described several ways for exchange of geodata using different XML applications. But how efficient are current databases in storing and managing such kind of datasets and are there methods to export database data using the mentioned XML data format?

4.1 Extended Relational Data Model

The ordinary *relational data model* is not really applicable to store geodata. As data is stored within tables, the only way to store coordinates is to define three columns (x,y,z) and store each point as one row entry. This might be practical as long as only single points have to be managed (e.g.: center of a city). Regarding the case, that an object is a polyline (e.g.: borderline of a community, center line of a street, pipeline, …), that schema is resulting in very large and complex coordinate lists, which have to be stored in rather long tables. Especially topographic datasets (e.g.: terrain models on a medium scale between 1:2.500 and 1:100.000), which mainly consist of point and line information, would be leading to very large data tables and cause a lack of performance.

In response to these reasons, a lot of development has been done within the past ten years in order to enhance the relational model with respect to the management of spatial data. It was not enough to adapt the structure of the data management; the query language SQL (Structured Query Language) had to be extended as well.

The I.P.F. developed TopDB (Topographic Database) about twelve years ago. This is a relational database, extended by the topological data types AREA, LINE, POINT and WINDOW as well as by topological operators such as INTERSECT. On top of this database, the application TopDM (Topographic Data Management) has been implemented, in order to provide an efficient access to the data, managed by TopDB. Other features are visualization and manipulation (editing) of the data [Hochstöger 1996]. In the meantime, all big database systems provide efficient data models to store and manage spatial data.

Most of these systems use the *object relational model*, similar to the one implemented in TopDB. That means, relational data types are extended by objects. If the coordinates of a geometry are defined as an object, it can be stored within one single entry in the table, instead of n-columns (where n is the dimension) with m rows (one per coordinate duple/triple).

As these commercial systems become more and more important an abstract interface to the application TopDM has been implemented in 2002 to allow for database access to TopDB and to Oracle using Spatial Extension as well. During this job it turned out that many problems and bugs do exist in this commercial, market leading software. The fact that the geodata market is still a niche market, compared to data warehousing or other database business solutions, might be a reason for that. There is a small light at the horizon named Location Based Services (LBS) which gains more and more importance on the database market for example to do logistic tasks in transportation management. So LBS are forced by concerns like Oracle.

However, most of the currently available commercial databases are designed for 'conventional' GIS-Solutions such as city administration, facility management, interactive mapping and so on. For topographic datasets, which differ in their structure (big amount of points per geometry, less attribute data), those data models are not optimal. There are often restrictions like a limited number of points per geometry (concerning laser scanner data, one dataset may consist of more than 100 million points) or problems to store the essential attributes within the given structure might occur. Therefore it is still important to force the research in this area!

4.2 Other Database Models

There is a big dominance of the relational model, but nevertheless there are several interesting developments of other models that try to gain more importance on the database market. Until now there is no truly object oriented database that has a considerable share of the market. But several big database companies, starting with Informix in 1997, enhanced their systems by objects as described before. This development is rather positive for spatial databases.

In 2000, the company 'lazy-Software' (http://www.lazysoft.com/) presented the associative model. The main advantage of this model is the independence between application and data structure. So the idea of reusability of applications can be realized in an easy way. If that model, which was developed with regards to the WWW, can take it's chance against the relational model will be seen within the following years. If it does, it has to be considered that quite some time will pass until there might be efficient enhancements for this model to be used on spatial data.

5 CONCLUSION AND FURTHER WORK

My intention was to present the key issues of XML related standards and technologies for geodata. During the past years, XML applications have been developed for different purposes, and many of them have been accepted by the industry for data exchange tasks in large scale business or data warehousing solutions.

As the geodata market can be described as niche market, the evolution in this area moves on slower. However, XML applications like GML and SVG have a high potential to become applicable formats for storage, exchange and especially presentation of geodata. For sure, their current state was implemented according to the OGC Simple Features Specification and many, especially performance considering problems, do occur on applying them on large data sets like topographic point clouds consisting of millions of points. But instead of ignoring those standards due to the described shortcomings, the development has to be forced to profit from the advantages of XML as a basis for data management, exchange and visualization within the geodata community.

6 LITERATURVERZEICHNIS

Dorninger P.: Aufbau eines globalen Bild- und Geländedatenarchivs des Planeten Mars unter Verwendung von XML Technologien, GeoNews, Software-Magazin für Vermessung und Geoinformation, Nr. 4/2002, November, p. 12-13.

Kraus K., Dorninger P.: Einrichtung eines Deponie-Informationssystems. Österreichische Zeitschrift für Vermessung und Geoinformation, VGI, 88. Jahrgang, Heft 2/ 2000,p. 144 - 145.

Gottlob G., Koch Ch., Reinhard P.: Efficient Algorithms for Processing XPath Queries, Proceedings of 28th International Conference on Very Large Data Bases (VLDB), Hong Kong, China, August 2002.

Hochstöger F.: Software for Managing Country-Wide Digital Elevation Data. International Archives for Photogrammetry and Remote Sensing, XXXI, Part B2, Vienna 1996, p. 160 - 163.

Mertz D.: XML and Compression, Exploring the entropy of documents, Gnosis Software, Inc, September 2001 http://www-106.ibm.com/developerworks/xml/library/x-matters13.html

OpenGIS Consortium (OGC): Geography Markup Language (GML) 2.0, OpenGIS® Implementation Specification, 20 February 2001, OGC Document Number: 01-029 http://www.opengis.net/gml/01-029/GML2.html

Reichardt M.: OGC's GML 2.0 – A New Wave of Open Geoprocessing on the Web, GeoInformtics, Magazine for Geo-IT Professionals, Issue July/August 2001, p. 18-21.

World Wide Web Consortium (W3C): Scalable Vector Graphics (SVG) 1.0 Specification, W3C Recommendation 04 September 2001. http://www.w3.org/TR/SVG/

All given links were accessed on 23rd of December 2002.