REAL-TIME GENERALIZATION OF GEODATA IN THE WEB

Lassi LEHTO, Tiina KILPELÄINEN
Finnish Geodetic Institute, Finland
Department of Cartography and Geoinformatics
Lassi.Lehto@fgi.fi, Tiina.Kilpelainen@fgi.fi

Working Group IV/5

KEY WORDS: Internet, Generalization, Data dissemination, Standards, GIS.

ABSTRACT

The growing demand for dissemination of spatial data in the Web gives at the same time new challenges also for the problem of generalization. The Extensible Markup Language (XML) technology is widely regarded as a basis for general data processing in the Web. The Geographic Information (GI) community has also started adopting XML techniques for GI processing on the Web platform. Once XML-based GI resources and services start proliferating in the Web, means are needed to transform the original data into an appropriate form for the given end user task and device. In this paper, the Extensible Stylesheet Language Transformation (XSLT) is introduced as a tool to perform real-time generalization of XML-encoded spatial data. Certain generalization effects can be achieved by filtering the source data according to pre-defined rules, by selecting a proper geometry from a set of alternative geometries, and by setting the style properties appropriately. The XSLT extension mechanism enables generalization methods based on sophisticated coordinate manipulations, e.g. line smoothing, to be performed. The paper concludes by presenting a three-tier pilot system, developed at the Finnish Geodetic Institute (FGI), consisting of a Smallworld database server, a Java Servlet middle layer and a Web browser-based visualization layer.

1 INTRODUCTION

The aim in the research presented in the article is in development of on-line generalization of spatial data for visualization on the Web environment. The Web-based visualization is a still growing means for dissemination of spatial data. The starting point for the research is a vision of the future map product as an *ad hoc* representation of the source dataset. The resulting temporary map display serves as a map in-demand, supporting the various individual needs of a casual map user. In the near future, the end user of spatial data will be provided with means to define the contents, spatial extent, layout and scale of the final map view. The direct access to geospatial data resources, available to an individual data user through network, emphasizes the need for sophisticated generalization mechanisms operational in real-time. The real-time (also called on-line or on-the-fly) generalization in the networked environment is becoming in the focus of interest of the research community.

The current map data delivery in the Internet is usually based on raster images. The spatial resolution of the image is decided on the server side, and a proper visualization of the map data is achieved only in that predefined resolution. Some map services provide maps in different scales, but the scale levels are often based on pre-created raster datasets. If the various scales are produced from a live database, then all the scale levels have to be stored and maintained in the database level beforehand. However, the approach does not allow an arbitrary display scale to be used. A service might support a continuous scale variation, but achieved simply by zooming the original map image without appropriate changes in the displayed map contents or in the generalization level of map objects and symbols. When using the real-time generalization approach the generalized datasets are not stored in the database but rather computed during the data request in real time. The use of individual *ad hoc* scales with appropriate content and generalization levels opens up more flexible use scenarios for geospatial databases. The need for management and maintenance of various generalized datasets is reduced once methods for reliable real-time generalization become available.

When using raster image-based map visualization the local interactivity and intelligence of the client application is also rather limited. A map service might provide the end user with a certain level of interactivity, however, usually with high latency and at the expense of a substantial increase in the network traffic. The various new standard developments related to vector-based processing of Web graphics generally, and spatial data in particular, provide promising solutions for development of more sophisticated client applications.

2 WEB TECHNOLOGIES

2.1 General

As the Web platform becomes the primary vehicle for data dissemination, the need to adopt the emerging Web standards and technologies in the network-based delivery of GI is evident. Some recent Web-related developments seem to facilitate more flexible and efficient use of existing GI resources (Kähkönen et al., 1999). In this paper we discuss the new data-driven processing model of the Web, enabled by the various Extensible Markup Language (XML) based techniques. Specifically, we concentrate on Extensible Stylesheet Language Transformations (XSLT), a new tool designed for carrying out transformations that involve considerable modifications in the order, structure and format of the source dataset.

2.2 Extensible Markup Language (XML)

One of the most significant Web standard developments currently is the activity of the World Wide Web Consortium (W3C) centered on the XML specification (W3C, 1998). The work on XML started around 1996 with a goal to develop a simplified Web-adapted version of the widely used Standard Generalized Markup Language (SGML), a language defining a markup syntax for structured text documents. XML is widely regarded as the next-generation encoding language for the Web. The specification is aimed at generic data transmission, not only at text document processing, as the popular HyperText Markup Language (HTML). The XML specification only deals with the logical structure and the exact syntax of the language, it does not fix the individual tags used in the markup. A rapid proliferation of XML vocabularies, adapted to various fields of society, has been a natural consequence of this flexibility. Thesevocabularies, once widely accepted and supported, enable the semantics of the data to be indicated, enabling the Web to become a vast repository of smart, self-describing data. The W3C Recommendation for the next version of the HTML specification has also been developed as an XML-application (XHTML) (W3C, 2000a).

XML specifies the constructs that can be used in building a structured text document. The most important construct is called *element*. An element is delimited by a start-tag and an end-tag. An element can have *attributes*, which are given inside its starting tag. The content of an element consists of text passages and/or of other elements. The tags, together with a few other constructs, are called the *markup*, which essentially describes the semantics and the logical structure of the document. The rest of the content of an XML document is called its *character data*. An XML document thus closely resembles a corresponding HTML code, although it introduces a few more strict syntax requirements. The extensive, semantically rich markup makes an XML file human-readable and self-describing. The structure of an XML document can naturally be expressed as a tree (see figure 1).

```
<Employee>
    <Name>
        <First>Lassi</First>
        <Last>Lehto</Last>
    </Name>
    <Email>Lassi.Lehto@fgi.fi</Email>
    <Organization>
        <Name>
            Finnish Geodetic Institute
        </Name>
        <Address>
            PO Box 15,
            FIN-02431 Masala
        </Address>
        <Country CountryCode="358">Finland</Country>
    </Organization>
</Employee>
```

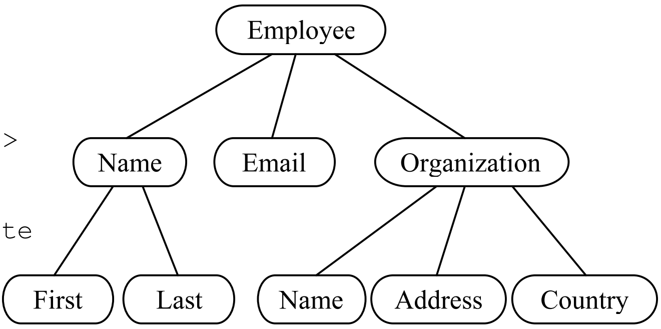


Figure 1. An example XML code snippet with the corresponding tree structure

An XML document that follows all the syntactic rules defined in the XML specification is said to be *well-formed.* A mechanism is provided in the XML specification to define detailed constraints on the structure of the document. These constraints are expressed as Document Type Definition (DTD) that specify the allowed element hierarchy and indicate the mandatory and optional attributes of the elements. Default values for the attributes can a lso be given. A document conforming to the rules defined in the DTD are said to be *valid*. The definitions in the DTD are expressed in a special-purpose, non-XML syntax (see figure 2).

```
      <!ELEMENT Employee (Name, Email, Organization)>
      <!ELEMENT Name (First, Last)>
      <!ELEMENT First (#PCDATA)>
      <!ELEMENT Last (#PCDATA)>
      <!ELEMENT Email (#PCDATA)>
      <!ELEMENT Organization (Name, Address, Country)>
      <!ELEMENT Name (#PCDATA)>
      <!ELEMENT Address (#PCDATA)>
      <!ELEMENT Country (#PCDATA)>
      <!ATTLIST Country CountryCode CDATA #REQUIRED>
```

Figure 2. An example DTD (corresponds to the dataset in figure 1)

Being an international cross-vendor specification, XML can be expected to gain support from many Web application developers, most notably from the makers of the popular Web browsers. Rendering facilities for XML content will thus soon become widely available. XML code is based purely on character encoding, making the code easy to create and edit, even with traditional text processing tools. Wide adoption of XML technology in the Web community ensures that all the basic tools needed to work with XML become available free of charge in the Web. One of the most useful sites to look for free XML tools is the recently founded Apache XML Project (Apache.org, 2000).

2.3 Extensible Stylesheet Language (XSL)

A basic principle in the design of the XML technology is the total separation of the contents of the data from its presentation characteristics. In the text document processing domain this principle forms a basis for the so-called multi-purpose publishing, an approach in which various presentations, aimed at different end user environments, are produced from a single source (Saarela, 1999). The Extensible Stylesheet Language (XSL) specification is being developed by the W3C as a tool for defining presentation characteristics of an XML dataset (W3C, 2000b). In connection to this work the W3C has created a specification for transforming XML documents, XSL Transformations (XSLT) (W3C, 1999a). XSLT is primarily designed for transforming XML documents for presentation purposes. Typical examples include dynamic creation of the table of contents, and creation of a tabular presentation of some data values in the source document. In the graphics domain XSLT could be used, for instance, to transform a dataset from an application-specific data structure to the new Web vector graphics standard Scalable Vector Graphics (SVG).

The XSL process is carried out by a special software module, called XSL stylesheet processor. The responsibilities and the expected behaviour of the XSL processor are well defined in the specification. An XSL process can be divided into two phases: tree transformation and formatting (see figure 3). The tree transformation is performed by an XSL processor. The input to the processor include a source tree and an XSL file containing the instructions that define the transformation. The results of the transformation are presented as a result tree, which contains all the presentation characteristics, in addition to the data contents selected by the transformation. Formatting phase is subsequently done by a formatter to produce the final output, in a form suitable for the given visualization environment. In the case of client application being able to interpret XML-code present in the result tree there is no need for a separate formatting phase. An XSL transformation is thus defined by setting up appropriate instructions in an XSL file; all the processing is carried out by an XSL processor and a formatter. There are several preliminary free XSL processor available in the Web, the most important of these being the XSL processor Xalan, available from Apache XML web site.

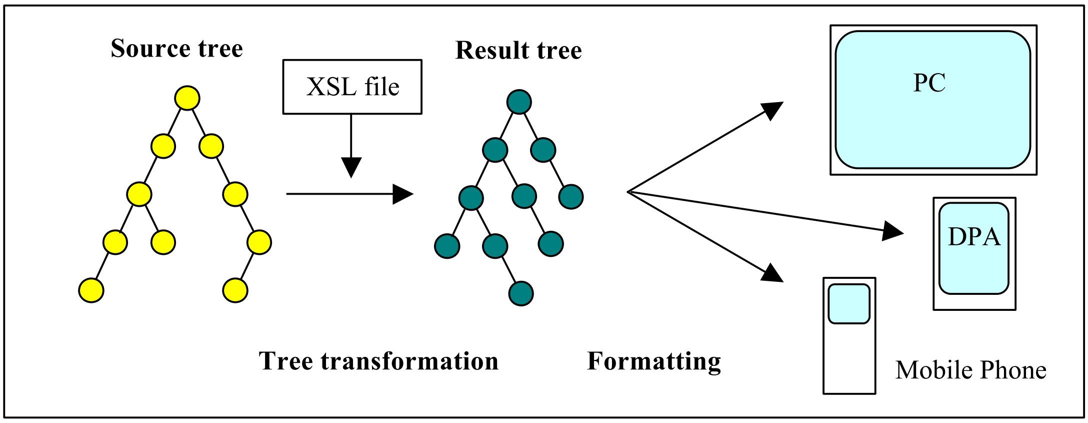


Figure 3. The two phases of the XSL process

3 GEOSPATIAL APPLICATIONS

3.1 XML as an Encoding Language for Spatial Data

XML has been proposed as a technology for processing structural data on the Web. New XML-based data models, called XML applications or XML vocabularies, are being developed for several application areas, the GI community not being an exception. At the same time the use of vector-based graphics is generally gaining importance in the Web. A notable example of this is the Scalable Vector Graphics (SVG), a new XML application for vector graphics being developed by the W3C.

DTDs have been widely used in presenting geospatial metadata. Examples include the Content Standard for Digital Geospatial Data (CSDGM) of the US Federal Geographic Data Committee (FGDC, 1998), and the Australian New Zealand Land Information Council's ANZMETA (ANZLIC, 1999). Recently XML has also been presented as a means to encode the spatial data itself, not only the metadata descriptions (Toon, 1999; Waters, 1999). The ISO TC211 has proposed XML as an encoding language for geospatial data (ISO TC211, 1999). Work is also going on at the Open GIS Consortium (OGC) to develop a spatial XML application, called Geography Markup Language (GML), based on the recently accepted OGC's Simple Features interoperability data model (OGC, 1999a). The use of XML-based encoding of spatial data in the Web will be a central issue in the next phase of the OGC's Web Mapping Testbed (OGC, 1999b) (Fletcher, 1999). Autodesk Inc. led industry consortium is also developing an XML application, called LandXML, for presenting spatial data related to land development processes (LandXML.org, 2000). The following is an example snippet from an XML-coded geospatial feature, reflecting ideas developed at the OGC (Lake, 1999).

```
<Feature identifier="1" typeName="building">
    <property name="Type">governmental</property>
    <geometryProperty typeName="extentOf">
        <Polygon srsName="4336">
            <outerBoundaryIs>
                <LinearRing>
                    <coordinates>
                    1.0,2.0 4.0,5.0 5.0,4.0 2.0,1.0
                    </coordinates>
                </LinearRing>
            </outerBoundaryIs>
        </Polygon>
    </geometryProperty>
</Feature>
```

Figure 4. An example snippet from a GML-encoded spatial Feature

XML provides natural support for the hierarchical data structures present in geospatial data. XML constructs can naturally be employed in many different ways to encode a spatial dataset, however, the standardization efforts of ISO TC211 and OGC are hoped to yield a common, internationally accepted XML vocabulary for geospatial use. A negative aspect in using XML for encoding geospatial data is the fact that XML is entirely based on character encoding. This encoding mechanism clearly does not provide the most efficient way to transmit predominantly numeric spatial data. An extensive markup notation may also degrade the overall performance of the process. For these reasons the XML-based spatial data encoding could be applied in transmissions of relatively small datasets only.

The standardization work of the W3C on the XML-based vector graphics will also provide useful tools for developers of map-related Web applications. SVG is a specification for a rather sophisticated vector graphics, expressible in XML syntax (W3C, 2000c). SVG is expected to become the *de facto* standard for vector graphics in the Web. Map information has been frequently mentioned as a typical application for SVG (Gould and Ribalaygua, 1999), and one of the most advanced SVG viewer prototypes is being developed in the context of geospatial data (CSIRO, 1999). The main benefits provided by SVG for map applications are: scalability (zoom in without a loss of precision), stylability (change of the styling parameters locally on the client side), searchability (place names inside the map can be searched for locally on the client) and linkability (individual map objects can be easily linked to external resources, e.g. to retrieve additional attribute information on request).

3.2 Geospatial Applications of XSL Transformation

Another XML technology providing interesting opportunities for processing of GI is XSL, a language for defining presentation characteristics of an XML dataset. Specially interesting for GI-related applications is the part of the XSL specification, called XSL Transformation (XSLT), which defines how an XML dataset can be transformed into another XML dataset. XSLT can be seen as a tool for transforming XML-based spatial data into some graphics format, for example SVG, for visualization purposes. Another task in which an XSL Transformation could be used is transforming XML-based GI datasets from one data model into another. The experiences gained at the FGI confirm that significant savings can be achieved if XSLT is used in data model conversion, instead of creating the conversion code from the scratch. XSLT provides also an interesting possibility to generalize the contents of spatial XML-encoded data, for instance to produce an appropriate representation for a given end user task or device. Lake has also presented XSLT as a tool for coordinate transformations and data integration (1999).

4 REAL-TIME GENERALIZATION BY XSLT

Use of XML as an encoding language for GI is obviously going to increase. This trend is evidenced by the development work done around GI-related XML standards like GML and LandXML, and by the interest GIS software developers and general database vendors have in building XML-ready solutions. Once XML-based GI services start appearing in the Web, tools will be needed for converting existing datasets into an appropriate form for a given application and visualizing device. In this context XSLT looks like a promising technology to perform data model conversions, styling operations, and real-time generalization of spatial data.

XSL Transformations are defined as templates that are matched against the source XML tree. For each matching source tree element a new element is created in the result tree by instantiating the contents of the template. The process continues recursively down the source tree until all the matching elements are transformed. The pattern expression in the match-attribute of the template follows a specific syntax defined in the XPath specification, a language for addressing internal parts of an XML document (W3C, 1999b). Selection operations based on element names or attribute values can easily be defined. Also the contents of an element can be tested in selection rules. XPath contains basic facilities for string, number and boolean computations, too. A few representative matching expressions are shown in the following XSL templates.

```
<xsl:template match="Feature[@typeName='Building']">
<xsl:template match="Feature[geometryProperty[@typeName=’centerOf’]]">
<xsl:template match="Feature[property[@typeName=’area’ and . &lt; 100]]">
<xsl:template
    match="FeatureCollection[@typeName=’Parcel’]/featureMember/Feature">
<xsl:template match="Feature[@typeName=’BorderStone’]">
    <xsl:if test="inside(geometryProperty/Point/coordinates,0,0,100,100)">
      ...
    </xsl:if>
```

Figure 5. XSL templates demonstrating different matching expressions of XPath

The given examples follow the GML vocabulary of the OGC. The first match expression would select all `Feature` elements having a `typeName` attribute with value ‘Building’. The second pattern would pick up all point geometries of type ‘`centerOf`’, the third all Features having a `property` element with typeName ‘area’ and the element contents being convertible to a number with value less than 100. The fourth expression would select all Features inside a `FeatureCollection` ‘Parcel’. The last one demonstrates an application-specific extension function that would test the contents of a `coordinates` element , for instance to select only the Features inside the given spatial extent for processing. The last example also shows how further conditions can be set by the `xsl:if` construct.

Certain generalization effects can be achieved when processing spatial data, by designing the templates appropriately. Data can be filtered in various different ways. The most obvious filtering operators are those based on object class (typeName attribute of a Feature element in the case of GML) or attribute (contents of a `property` element inside a

The following snippet from an XSLT-file would pick either the Polygon or the Point geometry of a building for inclusion in the result set depending on whether its type property be "governmental" or not (the use of alternative geometries naturally assumes that both are available in the Feature). The example also demonstrates a transformation from application-specific constructs (`Feature`,`property`,`Polygon` etc. of GML) into concepts of a general vector graphics language (`g`,`path` etc. of SVG). In the `Polygon` template an extension function is used for transforming the geographic coordinates of a GML Feature into the coordinate system of SVG. In the same extension function arbitrary generalization computations might be carried out. Simple generalization effects can further be achieved by properly setting the style properties of the result tree elements.

```
<xsl:template match="Feature[@typeName='building']">
    <xsl:if test="property[@typeName=’type’ and .="governmental"]>
        <g style="fill:red; stroke:black; stroke-width:2">
            <xsl:apply-templates select="geometryProperty/Polygon" />
        </g>
    </xsl:if>
    <xsl:if test="property[@typeName=’type’ and not(.="governmental")]>
        <g style="stroke:red; stroke-width:1">
            <xsl:apply-templates select="geometryProperty/Point" />
        </g>
    </xsl:if>
</xsl:template>

<xsl:template match="Polygon">
    <path d="coordTrans(outerBoundaryIs/linearRing/coordinates)" />
</xsl:template>
```

Figure 6. An example XSL Transformation involving geospatial data

5 PROTOTYPE DEVELOPMENT

A working prototype system has been developed at the Finnish Geodetic Institute (FGI) for testing the above-mentioned techniques. The prototype is based on a three-tier processing architecture (see figure 7). The first level consists of a Smallworld GIS database server. The middle tier has been developed in-house as a Java Servlet-based Web server extension. Communication between the Smallworld GIS and the middle tier is based on the CORBA technology and on an OGC Simple Features (SF) specification-compliant access interface provided by Smallworld. A client application on the third level has been built using a free map-visualization Java library, called OpenMap (BBN Technologies, 2000). Another client application tested is the SVG Viewer browser plugin from Adobe (Adobe, 2000).

Major part of the Java development of the prototype done at the FGI is related to the communication between the middle tier and the Smallworld Simple Features CORBA server and to the construction of the XML source tree from the received data. The source tree is constructed according to a proprietary data model, designed to facilitate easy XSLT transformations to the known destination data models. The middle tier makes use of some free XML tools available from Apache.org, like Xerces XML parser and Xalan XSLT processor.

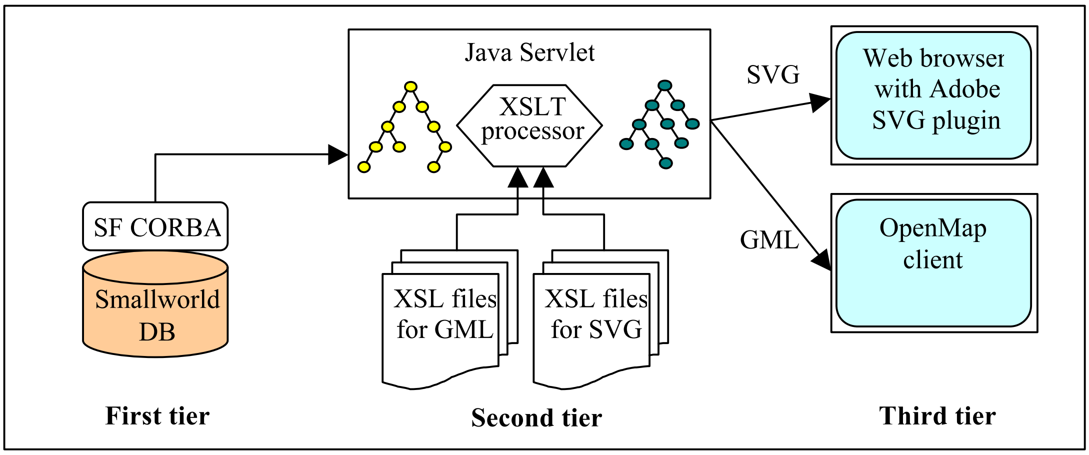
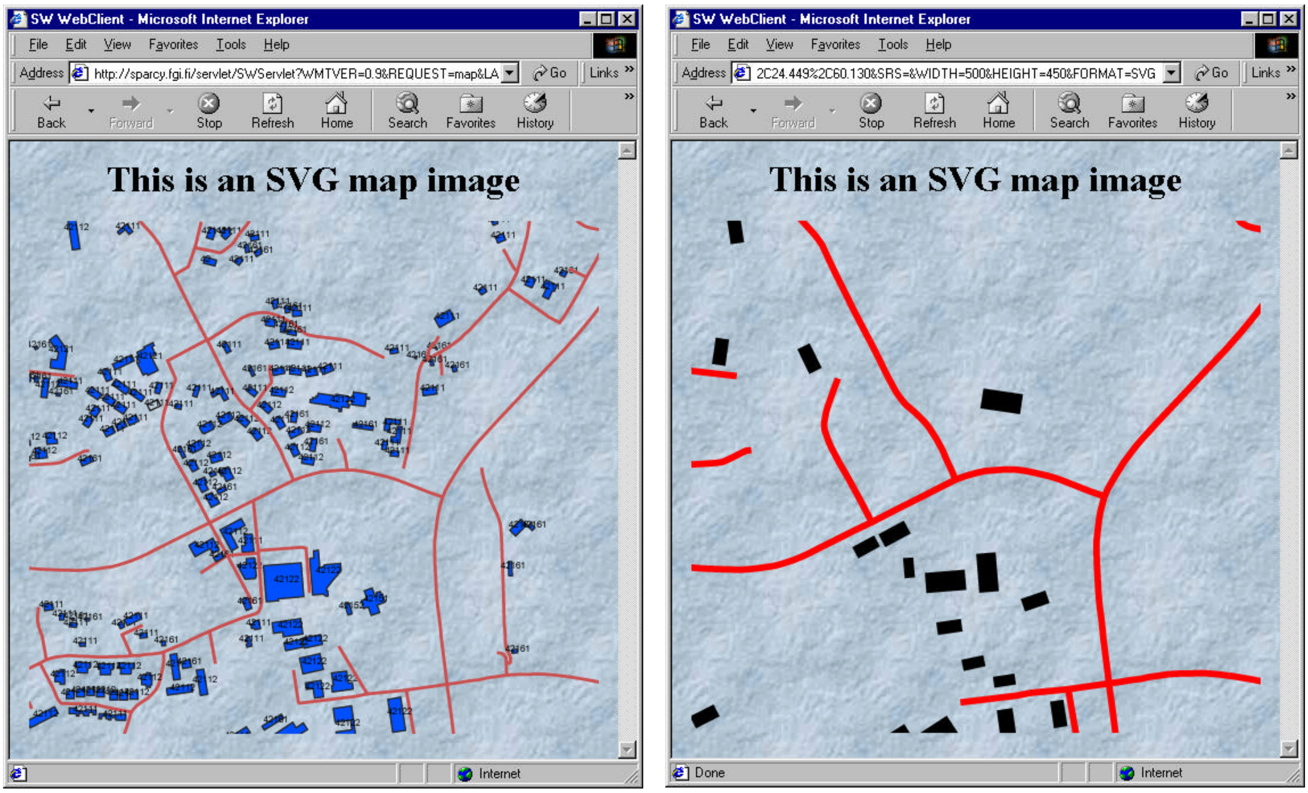


Figure 7. The pilot system's three -tier architecture

In the Servlet an XML tree is constructed from data retrieved from the Smallworld database. A proper XSL transformation is then applied to get the result tree in the desired XML vocabulary (GML and SVG are used), generalized appropriately for the end user device characteristics. As both encodings follow XML syntax, the result tree can be directly written out to the client, without a separate formatting procedure.

The emphasis of the intellectual work in building an XSLT-based service naturally concentrates on the design of appropriate XSLT files. The XPath expressions in the templates pick up the desired part of the source data for processing. The contents of the templates define the structure of the result tree. The XSLT files developed at the FGI provide a demonstrative example of the generalization results that can be achieved when transforming spatial data by an XSLT processor (see figure 8).



Address http://sparcy.fgi.fi/servlet/SWServlet?WMTVER=0.9&REQUEST=map&LA

Address 2C24.449%2C60.130&SRS=&WIDTH=500&HEIGHT=450&FORMAT=SVG

Figure 8. Two SVG displays of the same dataset, transformed by different XSLT processes. Two generalization methods are used in the process: selection and simplification.

6 CONCLUSION

As the XML-based data processing becomes widely adopted in the Web, solutions for GI delivery using this new technology are needed. Varying needs of the users of GI resources on the Web, and the diverse set of client devices they are using, call for mechanisms for generalizing geodatasets real-time. The XSLT specification is a promising solution to this need. Most simple generalization operations, like filtering out the unneeded parts of the dataset and selecting from among alternative geometries, are readily available. More sophisticated generalization tools can be added via the XSLT extension mechanism. Typical examples include coordinate manipulations, like line smoothing. The generalized datasets are written out as XML data and can thus be easily visualized in various XML-conscious client applications.

ACKNOWLEDGEMENTS

Test data for the project has been provided by the National Land Survey of Finland, which is gratefully acknowledged. This research has been supported by the Finnish Ministry of Agriculture and Forestry under grant number 5447/416/98 for the Finnish Geodetic Institute.

REFERENCES

Adobe, 2000. Scalable Vector Graphics. http://www.adobe.com/web/features/svg/main.html

ANZLIC, 1999. ANZMETA SGML/XML Document Type Definition (DTD) for geospatial metadata in Australasia. http://www.environment.gov.au/database/metadata/anzmeta/

Apache.org, 1999. Apache XML Project. http://xml.apache.org

BBN Technologies, 1999, OpenMap. http://openmap.bbn.com

CSIRO, 1999. SVG Viewer. http://sis.cmis.csiro.au/svg/

FGDC, 1998. Content Standard for Digital Geospatial Metadata (CSDGM). http://www.fgdc.gov/metadata/fgdc-std-001-1998.dtd

Fletcher S. J., 1999. Directions for the Geospatial Web, Mapping Awareness, September 1999.

Gould M., Ribalaygua A., 1999. A New Breed of Web-Enabled Graphics, GeoWorld, March 1999

ISO TC211, 1999. Geographic information - Part 18: Encoding, http://www.statkart.no/isotc211/wg4/welcome.html

Kähkönen J., Lehto L., Kilpeläinen T., Sarjakoski T., 1999. Interactive Visualisation of Geographical Objects on the Internet. Int. J. Geographical Information Science, 1999, Vol. 13, No. 4, pp. 429 -438.

Lake R., 1999. Xbed Technology Demonstrations. http://www.opengis.org/wmt/19990910_WMT_TECH_MAIN.ppt

LandXML.org, 2000. LandXML http://www.LandXML.org

OGC, 1999a. OpenGIS Simple Features Specification. http://www.opengis.org/techno/specs.htm

OGC, 1999b. Web Mapping Testbed. http://www.opengis.org/wmt/

Saarela, J., 1999. The role of metadata in electronic publishing.
Doctoral dissertation, Helsinki University of Technology, 1999.

Toon M., 1999. XML - a new language for the Web, Mapping Awareness, April 1999

W3C, 1998. Extensible Markup Language (XML) 1.0, W3C Recommendation 10-February-1998. http://www.w3.org/TR/1998/REC-xml-19980210/

W3C, 1999a. XSL Transformations (XSLT), Version 1.0, W3C Recommendation, 16 November 1999. http://www.w3.org/TR/xslt/

W3C, 1999b. XML Path Language (XPath), Version 1.0, W3C Recommendation, 16 November 1999. http://www.w3.org/TR/xpath/

W3C, 2000a. XHTML 1.0, The Extensible HyperText Markup Language, A Reformulation of HTML 4 in XML 1.0, W3C Recommendation 26 January 2000.
http://www.w3.org/TR/xhtml1/

W3C, 2000b. Extensible Stylesheet Language (XSL), Version 1.0, W3C Working Draft, 1 March 2000. http://www.w3.org/TR/xsl/

W3C, 2000c. Scalable Vector Graphics (SVG) 1.0 Specification W3C Working Draft, 03 March 2000. http://www.w3.org/TR/2000/03/WD-SVG-20000303/index.html

Waters N., 1999. Is XML the Answer for Internet-Based GIS?, GeoWorld, June 1999