

**Yayın Geliş Tarihi: 17.08.2018**

Cilt:4, Sayı:1, Yıl:2018, Sayfa 53 -69

Yayına Kabul Tarihi: 08.10.2018

ISSN: 2148-3752

Online Yayın Tarihi: 26.10.2018

VEB TABANLI CBS İÇİN XSLT İLE GML VERİLERİNİN GÖRSELLEŞTİRİLMESİ

Buket AKILLI

Murat KOMESLİ¹

Özet: Veb tabanlı Coğrafi Bilgi Sistemlerinin (CBS) geliştirilmesi, CBS endüstrisinde İnternet ve web teknolojilerinin gelişmesi ile gerçekleştirilmektedir. İnternette veb tabanlı CBS, tarayıcı yazılım üzerinde doğrudan çalışması, platformdan bağımsız olması ve ücretsiz kullanılabilir olması gibi birçok avantaj sunmaktadır. GIS işlevselliği sayesinde statik görüntüler veya dinamik karmaşık görüntüler gibi çeşitli görselleştirme imkânları sunmaktadır. Bunun yanı sıra, CBS'nin ortak kullanılabilmesi için, birlikte çalışabilirlik ve verimlilikten yoksundur. Bu sorunların üstesinden gelmek için, Coğrafi İşaretleme Dili (GML) olarak bilinen coğrafi grafik veri formatının kullanılması önerilmektedir. Coğrafi veriler GML ve Scalable Vector Graphics (SVG) ile yapılandırılabilir ve görselleştirilebilir. Web Feature Service (WFS), GML verilerini elde etmek için gerekli bir teknolojidir; coğrafi veriler, farklı Genişletilebilir Stil Sayfası Dil Dönüşümleri (XSLT) kullanılarak da görselleştirilebilir. Bu çalışmada, GML verileri önce mevcut WFS'den gerçek zamanlı olarak elde edilmiş ve daha sonra XSLT tarafından SVG'ye dönüştürülmüştür. Bu şekilde, coğrafi veriler ücretsiz tarayıcı yazılımlarında yüksek kaliteli vektör haritaları olarak görüntülenebilmektedir. Böylelikle verilerin birlikte çalışabilirliği ve dijital haritaların verimli yayılımı sağlanacaktır.

Anahtar kelimeler: Veri Görselleştirmesi, XSLT, GML, SVG.

VISUALIZATION OF GML DATA WITH XSLT FOR WEB BASED GIS

Abstract: Development of web-based Geographic Information Systems (GIS) is performed by popularity of Internet and web technologies in the GIS industry. Web-based GIS on the internet has many advantages like working directly on the browser, platform independent and being freely available. There are various kinds of visualization results like static images or dynamic complex images with GIS functionality. On the contrary, the common method of GIS lacks of interoperability and efficiency. To overcome these issues, the most common geo-graphical data format known as Geographic Markup Language (GML) is suggested to be utilized. Geo data can be structured and visualized with GML and Scalable Vector Graphics (SVG). Web Feature Service (WFS) is a required technology to obtain GML data, Geo data can also be visualized by using different Extensible Stylesheet Language Transformations (XSLT). In this work, GML data is first obtained from available WFS in real time and then is transformed into SVG by XSLT. In this way, geo data is visualized as high quality vector maps in free browser software. Data interoperability and efficient dissemination of digital maps are considered to be achieved.

Keywords: Data Visualization, XSLT, GML, SVG

¹ Contact Author: murat.komesli@yasar.edu.tr. Yaşar University, İzmir, Türkiye.

INTRODUCTION

GIS has a variety of different definitions and interpretations according to variety of different disciplines (Kapluhan, 2014). One of the definitions of GIS is “It is an information system that makes spatial-based information which is obtained with observations and non-spatial information, collecting, storing, presentations to the user as completeness”. In recent years, the internet has become popular rapidly in the area of Web-based GIS. Web-based GIS have started to become important areas of projects which are performed by software development corporations. The internet is preferred for the reason of ease of the presentation of the GIS and usage of this kind of information.

It is known that 80% of information is based on location information (Franklin, Carl and Paula Hane, April, 1992). Due to the importance of the spatial information, it becomes more important to visualize the surface of the Earth in a more real way. Additionally, there is a need to standardize geographical information which is utilized in software applications. These advances have started to be initialized by the corporations and have contributed to the establishment of interoperability.

GML has been developed as a solution for the interoperability problems by Open Geospatial Consortium (OGC). Presently, GML is used to transport the spatial/locational information among platforms and storing this information for future use. In GML documents, Geo data is stored as isolated from presentation elements because of XML based technology. With these properties, it is still possible to get different visualizations of a GML document on web browsers. GML data file’s visualization technique is actually the same as XML, since GML is a dialect of XML.

The interpretation of the XML document can be achieved with XSLT, and the visualization of the XML based document on a web browser can be achieved with Scalable Vector Graphics (SVG). GML data is visualized in SVG to provide high-performance by the XSLT translating mechanism (Herdy et al, 2008). SVG is utilized when developing user interface independently from resolution (Çelikbilek, 2011). At the same time, produced SVG documents can be styled with Cascading Style Sheets (CSS). Interactivity properties are made with JavaScript and ECMAScript scripting languages. Graphical representation is necessary for visualization of any location. The most preferable method/technology for visualization of GML based data is SVG technology. SVG technology constructs dynamic graphics. It manages to visualize spatial/geographical objects in an effective way. Since both are XML-based, GML and SVG have the same structure and format. For this reason, it is possible to get SVG vector graphics from GML data with XSLT transformation files. Additionally, it is also possible to visualize the same GML file with different stylesheet files.

This study is important since GML data were first obtained in real time from existing WFS and then converted to SVG by XSLT. In this way, geographic data can be displayed as high-quality vector maps in free browser software. Thus, the interoperability of data and the efficient dissemination of digital maps are ensured. The paper is structured as follows: after introducing the GML and XSLT, next part continues with the materials and methods including GML and XSLT. Then, related work and a comparison are mentioned. Next, application developed within this study is introduced. Finally, results and recommendations are presented.

MATERIALS AND METHODS

GEOGRAPHY MARKUP LANGUAGE (GML)

GML is based on XML and it is published by Open Geospatial Consortium (OGC). (opengeospatial.org, 2014) GML is developed for definition, storing, transporting of spatial/geographical data. In addition, it is developed for interoperability problems among heterogeneous GI systems. Geographical information is stored in GML format and is translated to any requested formats when it is required within web technologies. GML, which provides exchange of data between the various platforms, has been prepared according to the rules specified by OGC (OGC Standards, 2014).

GML was first modeled based on Resource Description Framework (RDF) modelling technique. RDF is a technology that defines the source of the World Wide Web (WWW). After that, OGC defines XML schemas inside the GML documents. So, it enables easy communication with a variety of geo databases (Jiang et. al, 2008).

GML is instantiated for web technologies and web based services. GML is prepared according to “Geographical Information Systems Specifications”. GML makes spatial data exchange possible among various platforms. A GML document contains spatial and non-spatial information which is related with spatial information. A GML document is used for not only GIS systems but also geographical database as a data file.

All the spatial details are obtained from consideration of the earth’s surface in abstract form. In positional detail, each of the details is defined as a set of names, types and value properties. Such as:

PropertyX: {name, type, value}

Together with the name of a type of a detail, the number of properties that it can have are indicated by the identification of the type. GML digitizes the real world and then gets two dimensional models of the real world. By achieving this, GML creates a collection of the details.

In the structure of GML, it includes in itself both a geometry element and properties of geometry elements for coding spatial information in the spatial data file. GML has many objects within documents to identify geographical features. These objects are “Coordinate Reference Systems”, “Geometry”, “Topology” ”Time”, “Units of Measure”, “Generalized Values”. In addition, there are basic geographical objects for 2D demonstration of real world. GML stores geographical information with it’s container. Some of these properties are named as “Basic Properties” of OGC. These are “Point”, “Multipoint”, “Polygon”, ”Multipolygon”, ”LineerRing”, “MultiGeometry”. They can be grouped in three groups which are “Point”, “Line” and “Polygon”. Table 1 shows the geometry elements in GML corresponding to SVG elements.

Table 1. SVG correspondences of GML Geometry Elements

GML	SVG
<Point>	It can be represented by a rectangle, a circle or an icon.<svg:point>, <svg:rect>

<LineString>	<svg:polyline>
<LinearRing>	<svg:polygon>
<Polygon>	<svg:polygon>
<Box>	<svg:rect>
<MultiPoint>	It can be represented by a rectangle, a circle or a icon.<svg:point>, <svg:rect>
<MultiLineString>	It is defined inside <g> element as a group of <svg:polyline> element.
<MultiPolygon>	It is defined inside <g> element as a group of <svg:polygon> element.

An important benefit of GML is that it retrieves the best resolution image related to the geographic area. At the same time, these maps which are translated to presentation format can be saved in the user's computer. Another benefit of GML is that it's being on web technology. It can be visualized with web browsers on the user's computer without any need for an additional software. In addition, it is different from raster maps. Content of maps which is obtained from a GML document can be seen by opening the documents and reading them on the fly. It is possible for GML documents to have different interpretation according to the different XSLT files. XSLT file can translate GML documents to the presentation language.

Update and insert operations can be done easily on the GML file by compiling it in a typical word processing application. Another property is that GML files can hold isolated both geographical/spatial information and non-spatial information related with spatial information in the same file. Visualizing this information can be linked to related locations with the URL. And then, it can be displayed with all of the related information when clicked on the geographical feature on the user's computer. In addition, it is possible to filter operations in the geographical information. This operation is done by the querying of non-spatial information. Users have the opportunity to select themes for filtering results. These features are just as related to and worked on the GML file. On the SVG documents which are obtained from GML documents, it is possible to work with JavaScript language by adding animation. GML is a spatial data format to translate data to each other. The obtained benefit of using GML technology is that it is suitable for all usages as a unique format.

There are many properties in GML which must be defined in the schema document in order to obtain a valid GML document. There is no special style information in a GML document. For this reason, if it is required to visualize GML documents on the web browser, it is necessary to translate the document to a presentation format which is suitable for display. This technology is XSLT which works on a web platform.

EXTENSIBLE STYLESHEET LANGUAGE TRANSFORMATION (XSLT)

XSLT is a language that is primarily designed for transforming one XML document into another (Clark, 1999). XSLT is a language which provides a mechanism to transform and manipulate XML data. It is a World Wide Web Consortium (W3C) recommendation (W3C-XSLT, 2014). Extensible Stylesheet Language (XSL) is developed by W3C since there was a need to overcome for an XML-based Stylesheet

Language. By means of XSLT, elements and attributes can be added/modified/removed to/from the output file. Elements can also be re-arranged and sorted. Tests can be performed to make decisions about which elements to hide and display, and so forth. XSLT is based on Document Style Semantics and Speciation Language (DSSSL). Since XSLT language has a small part of XSL, it can be converted to XML document which isn't in a file format before it is defined. Thus, there is no information about XML document defined beforehand. So, it is not possible to determine the visualization of which form of the document will result. For this reason, there is a need for an additional document which helps XML documents in supplying information and visualizing them. This additional document is an XSL document which has XSL specifications. These specifications are developed by the W3C. Specifications are a language that defines which way should be defined for visualization of SVG document. XSL is a recommended tool for producing SVG. There are three sections in XSL structure. These are XSLT, XML Path Language (XPath) and XSL-FO (XSL Formatting Objects). XSLT translate GML documents into SVG documents. XPath is an addressing language in order to search in GML documents. On the other hand XSL-FO formats GML documents. XSLT is constructed by combining XSL-FO XSL technologies as stated below:

$$\text{XSL} = \text{XSLT} + \text{XSLFO} + \text{XPath}$$

XSL is designed for adding style to XML documents by using Formatting Objects (FO). Objects named as FO are used for identifying structures in the XML pages. XSLT represents the important parts of XSL Standards. XSLT translates XML documents to other type of documents. XSLT is a pre-defined language translation technology. During the translation process, the target document is created without changing the source document. In this study, GML document is the source document and SVG document is the target document. Hence, GML is translated to SVG document. There is no meta information about meaning of the GML document exists inside it. For this reason, it is needed to translate the geo-spatial data in GML into a viewable format by means of XSLT translation technology. In GML, data and presentation of data are separated from each other. In XSLT style files, there is a set of templates which contain many rules. Thanks to these templates GML content can be displayed as a map within a web page. XSLT files are created with an XML-based template language. One of the biggest advantages of the working with XSLT translating technology is that it is possible to change on the document easily. So, it is not necessary to compile the entire project. Another advantage is that it is suitable for different comments with different XSLT stylesheet files.

XML is not a programming language. It is a mark-up language to store, model and transport the data. The programming is normally performed by coding XSLT. Like any other programming language, XSLT also contains programming language components. These are "Data Types (Boolean, number, external type definitions)", "Operations (<xsl:template>, <xsl:apply-templates>, <xsl:match>, <xsl:sort>, <xsl:output> etc.)", "Flow control mechanisms (<xsl:if>, <xsl:for-each>, <xsl:choose> etc.)" and other programming language structures definitions. The first noticeable advantage of the XSLT is that it is an Object-Oriented language. It is platform independent. Changes are made independently from projects. It has an interface which is standardized for positional operations.

XSLT documents should be prepared as well-formed and complete form for translating process. At the same time, it should implement specification rules. XSLT file translates every node of the XML elements into SVG elements. When XSLT is traversed on its tree, it uses XPath expressions to access the different parts of the tree (Kılınç, 2014).

The structure of the XSLT document is just like the structure of an XML document. Below is an example of the structure.

```
<?xml version="1.0" encoding="ISO-8859-1">
  <xsl:stylesheet                                version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    </xsl:stylesheet>
```

XPath is a language which is used for referencing of selected special section of GML documents and getting around according to these references. It is like SQL (Structured Query Language). XPath helps to find data, in other words, it helps to reach elements and attribute values in desired characteristics in the documents. XPath makes necessary definition for selecting the node to be included into a loop. It is known that XPath can create nodes lists. It can also produce new values by using a set of functions. Also, it can both make comparisons and identify every node of path in XPath and XML documents. During this process, node list can be generated by using a query operation. In this language, syntax is used like an extensible text-based path file. Path file syntax is a path which is obtained by addressing of the tree's special sections. In other words, XPath is an assistive technology for defining a pattern. Some operators are needed to make a transaction by the Xpath. In the below code, there is a template which is prepared in XSL for "gml:Polygon" element located in a GML document.

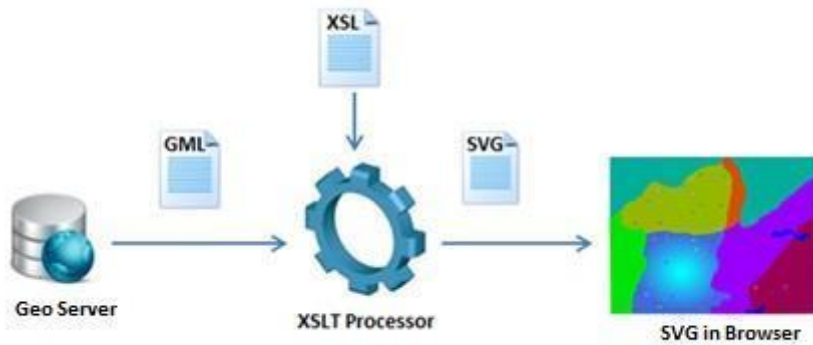
```
<xsl:template match="gml:Polygon">
  <xsl:variable name="ID"><xsl:value-of select="@gml:id"/></xsl:variable>
  <xsl:if
  test="contains(//gml:Polygon/gml:outerBoundaryIs/gml:LinearRing/gml:coordinates,')">
    <xsl:variable name="clist">
      <xsl:value-of
      select="//gml:outerBoundaryIs/gml:LinearRing/gml:coordinates"/>
    </xsl:variable>
    <polygon id="{ID}" class="polyline" points="{Clist}" class="polyline"
    style="fill:red;stroke:blue;stroke-width:5"></polygon>
    <polyline id="{ID}" points="{Clist}" class="polyline"/>
    <set attributeName="opacity" from="1" to="0.5" begin="mouseover"
    end="mouseout"/>
  </xsl:if>
  <path id="{ID}" d="{Clist}" style="fill:#D9F7FF;stroke:black;stroke-
  width:1;" />
```

```
</xsl:if>
</xsl:template>
```

In the code, “polygon” element in GML document is translated to “Polygon”, “Polyline” or “Path” element in SVG document. Coordinate values in the “Polygon” element are reflected into the “Polygon”, “Polyline” or a “Path” element. With this template, it is possible to handle all “Polygon” elements in the GML document as a “Polygon”, a “Polyline” or a “Path” element in the SVG document. In other words, the template is applied to the GML document and then the SVG elements are produced. Like this template, there are many other templates for translating the GML elements to SVG elements.

The working mechanism of the XSLT processors is seen in Figure 1. In the figure, all components are considered in a tree structure. GML, SVG and XSLT tree structures are working together. XSLT processor translates GML tree structure into SVG tree structure. During the translation processes, the XSLT tree structure works actively. In the processors, the mapping process is executed. The mapping process is defined in the XSLT templates. If mapping is successful, SVG codes which are located in the template are included in the output document. In Figure 1, there is a showcase of the process.

Figure 1. Translating of GML to SVG



An example utilization of this technique can be done to produce SVG files from GML/XML documents by using Saxon® executable. Saxon v9 (saxon.sourceforge.net, 2014) is used to process XML documents. It is particularly used for transforming XML data into other formats. Some features of Saxon are listed below:

- An XSLT processor, which implements the Version 1.0 XSLT and XPath Recommendations from the World Wide Web Consortium,
- A Java library, which supports a similar processing model to XSL, but allows full programming capability,
- An improved version of the Ælfred parser that is written by David Megginson, then of Microstar.

In this study, Saxon v9 is used for generating SVG files by using GML and XSLT documents.

RELATED WORKS

(Taladoire, 2001) has been referred to be displayed with the GML open standard. It is mentioned that GML files, XML schemas and XSLT technologies and works are realized by these technologies.

(Tennakoon, 2003) focused the interoperability of the applications and on visualizing GML data with XSLT technology and as related to this topic, visualization of GML data with the provided XSLT. During the process of displaying, CSS technology and client-side coding method also took place.

(Bonati, 2003) is mentioned that advantages of using GML data format in “SVG Explorer of GML Data” named work. It is also referred that it is possible that display through XSLT with SVG. It is described to acquisition process of GML data with WFS (Web Feature Service) services.

An application which is developed in 2003, named as “Publishing GML data as interactive SVG maps”. This application display GML data as SVG format. Work is realized and developed by software specialist Alison Meynert. Technologies which are located in the development of technologies for the application are XSLTC application which belongs to Apache and Java. Translation of GML data to SVG is realized with java codes. In the XSLT stylesheet files, was used, with XSLT extension functions (Saxonica Extended Functions, 2014). Within the application, GML data converted to SVG format dynamically, on the fly. (Meynert, 2003)

Another research work, in Finland, a work titled with “Schema Translations by XSLT for GML-Encoded Geospatial Data in Heterogeneous Web-Service Environment”. The work was done in 2004. As a solution of interoperability, it is pointed out that visualization of GML data with XSLT schema in heterogeneous web services. (Lehto and Sarjakoski, 2004).

Another survey conducted in China in 2004, is named as “Modeling GML and SVG Data for Web GIS”. The survey was handled by Liang ZOU. In the parallel to this survey, it tried visualization of GML data. (Liang, 2004)

In 2006, a work which is researched as master thesis, “GML processing with XSLT and spatial extensions”; the aim was to write a GeoXSLT that attempted to translate data files which contain geographical information encoded with GML. (Fredrik and Klausen, 2003).

The work conducted within this paper differs from the related works that GML data was first obtained in real time from existing WFS and then converted to SVG by XSLT. In this way, geographic data can be displayed as high-quality vector maps in free browser software. Thus, the interoperability of data and the efficient dissemination of digital maps is ensured.

APPLICATION

In this study two types of applications are implemented. First one, shown in the Figure 2, is a demonstration of creating a GML document with personalized tags, and personalized XSLT translation tags. A GML representation of a typical city plan is generated first (See Appendix-A), and based on that GML, an XSLT document is created (See Appendix-B). SVG file is generated by using Saxon Home Edition on

the basis of GML data and XSLT. As shown on Figure 2, with the styling engine, there is an SVG file generated which is a graphical output on browser. Below is the SVG output of the first part of the study.

Figure 2. Utilizing Saxon

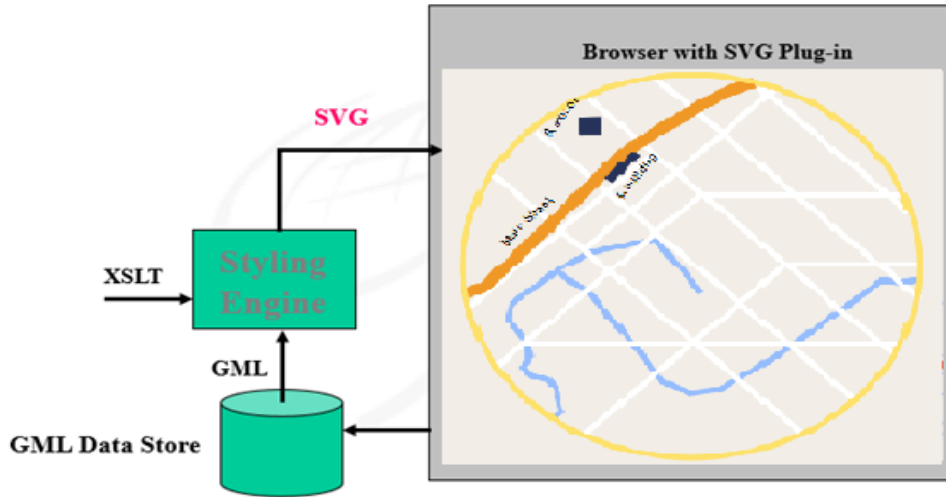
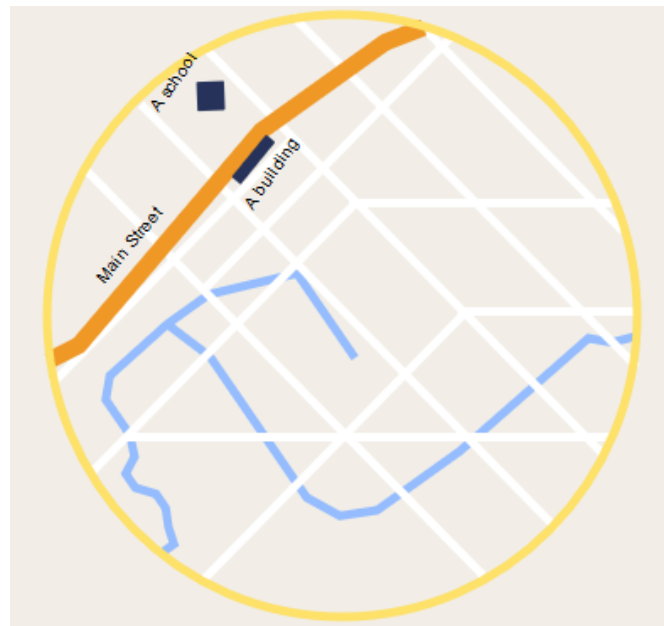


Figure 3. Produced SVG File



The second part of the study, shown in Figure 4, is aimed to create a styling engine instead of using external XSLT processors, such as Saxon.

The programming language used in this study is C# on the .Net platform. In this research's application, Microsoft XML technology was preferred. This part of the study has three sections. One of the sections is Web Feature Service (WFS) section. In section two, it is processed with Web Map Service (WMS) files which

are obtain from “Workspace Manager” application. In last section, there is a GML viewer to visualize GML files.

Figure 4. The Application Homepage



WFS CLIENT SECTION

In this section, WFS server connection is made by the user. It is developed with using Mapxtreme development kit for downloading WFS services. If it is wanted, other new connections can add the system. After connecting the WFS server, GML data is obtained by the connection. This GML data is saved on the C directory. This GML data needs to manipulating process. After manipulating process it become available for interpreting. With using prepared general XSLT, it is possible to display text based GML data to two dimensional Scalable Vector Graphics (SVG) format.

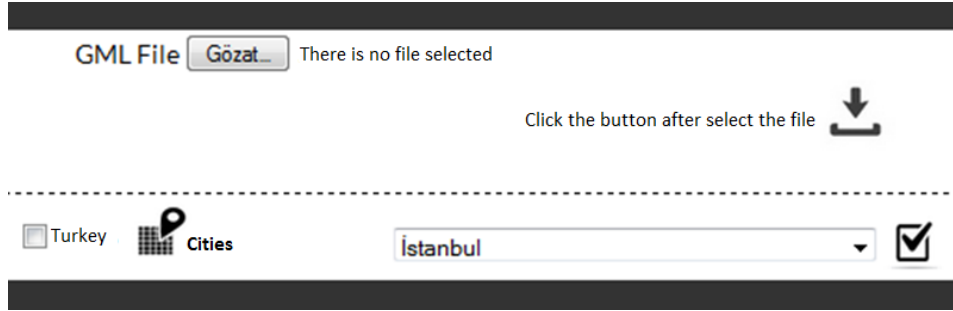
MAPINFO TAB FILES SECTION

This application is related to Mapinfo application. “Workspace Manager” translates TAB MapInfo® files to .wms file.

GML VIEWER SECTION

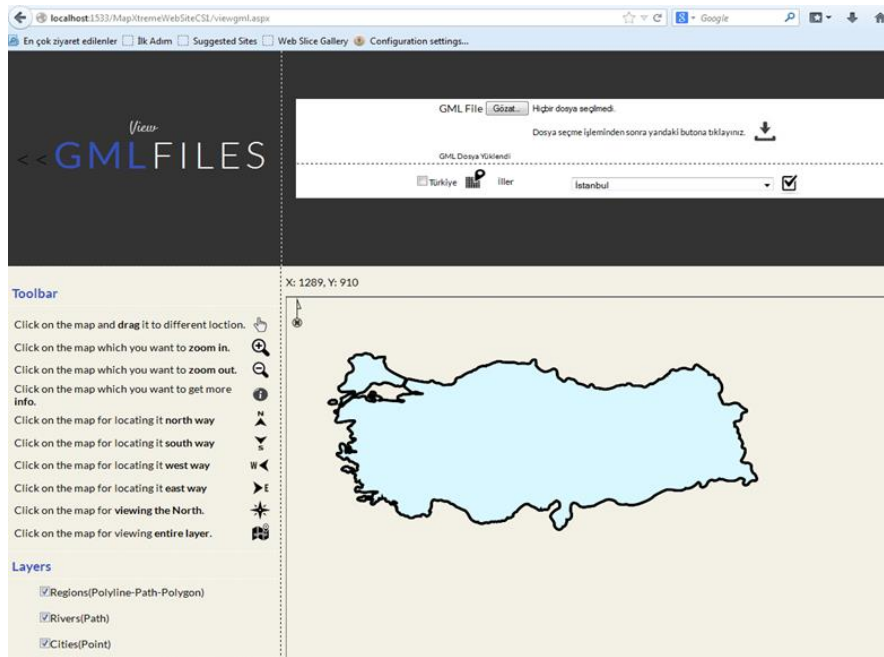
GML files which are on the local machine are visualized by the section. All explained sections are shown at Figure 5. In the last tab, there is information about the project. We will examine those sections respectively. If it is want, as in the screenshot below, a GML file with FileLoader can be selected.

Figure 5. GML File Load Panel



A file which is selected from the “Gözet” button is for browse purposes included with clicking the button option. Figure 6 displays the polygon image is which belongs to Turkey.

Figure 6. GML Visualization of Turkey



With the help of the "Toolbar" which is on the left side, it is possible to make various kinds of visualizing operations (zooming, floating, viewing entire layer) on the displayed SVG elements. In “Layers” section it can be provided to show and hide of the geometry elements.

RESULTS AND RECOMMENDATIONS

Because GML does not provide any information about how the map data should be displayed, visualization of GML data is important especially for web-based GIS. GML data visualization technology is still in developing phase (Lin, 2006). Currently, version 3.3 has been published (opengeospatial.org, 2018). There are various kinds of technologies and numerous solutions are provided from different approach. It is believed that the SVG technology is a better choice for GML data

visualization. In this study, XSLT based GML visualization strategy is used. It is seen that the SVG technology which is produced from XSLT technology is a relatively excellent on GML data visualization. There are advantages of the approach. First of all, maps which are displayed as SVG format can be stored and saved on user's computer. In addition, it is possible to make change on the SVG objects and their properties by the user side. Secondly, thanks to the Web browsers, it is not necessary any extra imaging program for visualization of GML data. Another advantage is that it is possible to store spatial and non-spatial information which is related to spatial information in the same file. It is possible to two-dimensionally display thanks to the use of the XSLT technology. At the same time, there is an advantage is that GML files can be displayed in different formats according to the documentation of the different conversion. It can be listed results of using SVG technology. Firstly, it is an open standard since it is developed by W3C. Another benefit is that it has widespread support by web industry leaders. Also it is editable, zoom capable.

This study can be applied to wireless PDA, car navigation systems, and mobile phones. In addition, web services like Web Feature Service, Web Map Service and Web Coverage Services can be developed to utilize the GML data.

ACKNOWLEDGEMENT

The authors acknowledge Tuğçe Yılmaz and Onur Ağbulut for their supports.

REFERENCES

Aras, İbrahim and Yıldız, Ferruh (2011). "İnternet Tabanlı CBS'nin Sivil ve Askerî Amaçlı Acil Durum Uygulamalarında Kullanılmasında Yeni Bir Yaklaşım", Ankara.

Bonati, Luca Piazza; Fortunati, Luciano; Fresta, Giuseppe (2003). "SVG Explorer of GML Data", link:"<http://www.svgopen.org/2003/papers/SvgExplorerOfGmlData>".

Clark, James (1999). XSL Transformations (XSLT) Version 1.0. W3C Recommendation. Copyright 1999 World Wide Web Consortium. URL: <http://www.w3c.org/TR/xslt/>.

Çelikkilek, İbrahim (2011). "Her Yönüyle HTML 5", Kodlab Yayıncılık, Sultanahmet-İstanbul.

Dempsey, Caitlin (2001). "Geography Markup Language (GML) 2.0 – Enabling the Geospatial Web", <http://www.gislounge.com/geography-markup-language-gml-20-enabling-the-geo-spatial-web/>, (Access date: June 2018).

Franklin, Carl and Paula Hane, "An introduction to GIS: linking maps to databases," Database. 15 (2) April, 1992, 17-22. (Access date: December 2014).

Fredrik, Carl; Klausen, Melsom (2006). "GML processing with XSLT and spatial extensions" URL:"<http://www.svgopen.org/2003/papers/SvgExplorerOfGmlData>".

Herdy, Kenneth S.; Burggraf, David S.and Cameron, Robert D. (2014). URL:

“http://www.svgopen.com/2008/papers/74HighPerformance_GML_to_Transform_for_the_Visual_Presentation_of_Geographic_Data_in_WebBased_Mapping_Systems/#LuSantosSripadaKou07”, (Access date: June 2018).

Jiang Jun, Yang Chong-jun, Ren Ying-chao, Jiang Miao (2008). “The Research and Application of WFS Based GML”.

Kapluhan, Erol (2014). “The Importance and Necessity of the Use of Geographic Information System (GIS) In Geography Teaching”, Marmara Coğrafya Dergisi Sayı: 29, Ocak-2014, S.34-59.

Kılınc, Deniz ve Prof Dr. Alp Kut (2014). DOI: 10.13140/2.1.3165.8248 Conference: Akademik Bilişim 04, Karadeniz Teknik Üniversitesi, Trabzon.

Komesli, Murat and Ünalır, Murat Osman (2004). Coğrafi Bilgi Sistemlerinde Coğrafi Verilerin GML (Geographic Markup Language) ile Modellenmesi, Bornova/İzmir.

Liang, Zou (2004). ”Modeling GML and SVG Data for Web GIS”. Pekin University, China.

Lin, Zhang; Shi-Bin, Chen (2006). “Study of GML-Based Geographical Data Visualization Strategy”, Proceedings of the 10th WSEAS International Conference on Computers, Vouliagmeni, Athens, Greece, (pp138-142),.

Lehto, L. and Sarjakoski, T. (2004). “Schema Translations by XSLT for GML encoded Geospatial Data in Heterogenous Web-Service Environment”.

Meynert, Alison (2003). “Publishing GML data as interactive SVG maps”, URL: ”<http://www.svgopen.org/2003/papers/cleopatra/>”.

OGC (2014). Geography Markup Language (GML) Standarts, <http://www.opengeospatial.org/standarts/gml> (Access date: July 2018).

Taladoire, Gilles (2001). “Geospatial Data Integration and Visualization Using Open Standard”.

Tennakoon (2003). “Visualization of GML data using XSLT”, ITC, Netherlands.

Wikipedia (2010). “Coğrafi Bilgi Standartları”, URL:“http://tr.wikipedia.org/wiki/Co%C4%9Frafi_bilgi_standartlar%C4%B1” (Access date: June 2018).

W3C XSLT (2014). URL: <http://www.w3.org/TR/xslt> (Access date: June 2018).

<http://www.opengeospatial.org/standards/gml> (Access date: December 2017).

<http://saxon.sourceforge.net/> (Access date: December 2017).

<http://www.saxonica.com/welcome/welcome.xml> (Access date: December 2017).

APPENDICES

APPENDIX-A


```

    <xsl:for-each select="Street/gml:coord">
      <xsl:variable name="gx" select="gml:x" />
      <xsl:variable name="gy" select="gml:y" />
      <xsl:variable
        name="gheight"
        select="gml:height" />
      <rect x="{ $gx }" y="{ $gy }" height="{ $gheight }"
        width="5" style="fill:{ @fill };" />
    </xsl:for-each>
  </g>
  <g id="road" transform="translate(20,250) rotate(45
  100 100)">
    <xsl:for-each select="road/gml:coord">
      <xsl:variable name="gx" select="gml:x" />
      <xsl:variable name="gy" select="gml:y" />
      <xsl:variable
        name="gheight"
        select="gml:height" />
      <rect x="{ $gx }" y="{ $gy }" height="{ $gheight }"
        width="5" style="fill:{ @fill };" />
    </xsl:for-each>
  </g>
  <g id="intersectRoad" transform="translate(20,250)
  rotate(90 100 100)">
    <xsl:for-each select="intersectRoad/gml:coord">
      <xsl:variable name="gx" select="gml:x" />
      <xsl:variable name="gy" select="gml:y" />
      <xsl:variable
        name="gheight"
        select="gml:height" />
      <rect x="{ $gx }" y="{ $gy }" height="{ $gheight }"
        width="5" style="fill:{ @fill };" />
    </xsl:for-each>
  </g>
  <g id="highway" transform="translate(20,100)
  rotate(-125 100 100)">
    <xsl:for-each select="highway/gml:coord">
      <xsl:variable name="points" select="gml:points"
      />
      <polyline
        points="{ $points }"
        style="fill:none;stroke:rgb(239,152,37);stroke-width:10"
      />
      <text x="{ position()*25 + 7.5 }" y="90"
        style="font-size:10px;font-family:arial;text-
        anchor:left;baseline-shift:-15;" transform="translate(-
        12,40) rotate(75 100 100)">
        <xsl:value-of select="gml:y"/>
      </text>
    </xsl:for-each>
  </g>
  <g id="building" transform="translate(-91,100)
  rotate(-61 125 61)">

```



```

        <xsl:for-each select="building/gml:coord">
            <xsl:variable name="points" select="gml:points"
/>
            <polyline                points="{ $points }"
style="fill:none;stroke:rgb(38,50,89);stroke-width:7" />
            <text      x="{ position()*200-30}"      y="208"
style="font-size:10px;font-family:arial;text-
anchor:left;baseline-shift:-150;" transform="rotate(6 100
100)">
                <xsl:value-of select="gml:y"/>
            </text>
        </xsl:for-each>
    </g>
    <g id="circleway">
        <xsl:for-each select="circleway/gml:coord">
            <xsl:variable name="cx" select="gml:x" />
            <xsl:variable name="cy" select="gml:y" />
            <xsl:variable name="r" select="gml:r" />
            <circle  cx="{ $cx }"  cy="{ $cy }"  fill="none"
r="{ $r }" stroke="#FEE168" stroke-width="5" />
        </xsl:for-each>
    </g>
    <g id="campus" transform="translate(-91,100)
rotate(-61 125 61)">
        <xsl:for-each select="campus/gml:coord">
            <xsl:variable name="points" select="gml:points"
/>
            <polyline                points="{ $points }"
style="fill:none;stroke:rgb(38,50,89);stroke-width:17" />
            <text      x="{ position()*200-30}"      y="130"
style="font-size:10px;font-family:arial;text-
anchor:left;baseline-shift:-150;" transform="rotate(6 100
100)">
                <xsl:value-of select="gml:y"/>
            </text>
        </xsl:for-each>
    </g>
</svg>
</xsl:template>
</xsl:transform>

```