# GML – A Real Standard?

**Izabela ŚLIŻ and Piotr CICHOCIŃSKI, Poland**

**Key words**: Geography Markup Language, import

## SUMMARY

The rapid development of Geographical Information Systems (GIS) in the nineties of 20th century, resulted in creation of many new programs, which make possible building spatial databases, where each of them offered his own different file format. Along with internet spread, there became opportunities and needs of data exchange between these databases. To make effective information transfer possible, it is essential to introduce unified spatial data formats. Several industry standards exist but there was a lack of formalized and universal one. In 2001 Open Geospatial Consortium (OGC – the organization associating main GIS software producers) accepted Geography Markup Language (GML) as the standard for coding, spreading and collecting geographical information. Despite its formal existence for several years, it did not gain the proper place among GIS software users, although there are several programs advertising their GML handling.

The aim of the work presented in this paper, was to examine whether available GIS software has the ability to load spatial data in this format and whether it does this properly. After short description of GML file structure, there were in detail analyzed capabilities of selected GIS programs in the area of spatial data import. Both commercial (ArcGIS) and free software (OpenJUMP, Quantum GIS) were considered.

As an example, the excerpt from Topographical Database (TBD) was used, recorded in GML in accordance with specification published as technical guidelines by Surveyor General of Poland. This is the only formal GML usage in Poland as yet. For comparison there were also used, available on the internet, exemplary files coming from AAA-NAS (Germany), Ordnance Survey MasterMap (Great Britain) and TOP10NL (Netherlands).

Conducted research showed, that majority of programs refused to import specific TBD GML files. It seems that the main problem is the proper schema file reading. That is why special attention deserves, available in OpenJUMP, possibility of making simplified schema, but it requires knowledge of imported GML file data structure and knowledge of GML itself.

# GML – A Real Standard?

**Izabela ŚLIŻ and Piotr CICHOCIŃSKI, Poland**

## 1. INTRODUCTION

The rapid development of Geographical Information Systems (GIS) in the nineties of 20th century, resulted in creation of many new programs which make possible building spatial databases, where each of them offered his own different file format. Along with internet spread, there became opportunities and needs of data exchange between these databases. To make effective information transfer possible, it is essential to introduce unified spatial data formats. Several industry standards exist but there was a lack of formalized and universal one. In 2001 Open Geospatial Consortium (OGC – the organization associating main GIS software producers) accepted Geography Markup Language (GML) as the standard for coding, spreading and collecting geographical information. Recently it even became an ISO standard (ISO 19136:2007). GML is the implementation of XML (eXtensible Markup Language) (Cichociński 2001). XML is a formal markup language, which can be used to carry information between computer systems. Its main application area is recording richly structured documentation. It is pretty straightforward to create and supervise complex hierarchical data structures using XML. Such structures are often met in geographic applications. XML does not define neither the semantics nor the set of tags. As a matter of fact XML is the language of data description or more correctly metalanguage used for expression of data description languages (markup languages). Because it is extensible, practically any user who wants to record his data can do it according to his needs. The only requirement is to attach schema files to files containing data, allowing proper data "decoding". Schema is a model of information structure description. It is a term borrowed from the world of databases, where it describes data structures in relational tables. In the context of XML, schema describes model for the whole document class. This model describes allowed positions of tags and text in structurally correct document. A schema can be also seen as an agreement of common vocabulary for particular purpose requiring document interchange.

In schemas, models are described using rules. A rule defines, what can occur in a given context. Basically two types of rules exist: content model rules describe the order of element occurrences, and data type rules describe valid data units. Using schemas authors are able to define structure of their documents and allowed data types. Subsequently parsing software can check the conformity of documents being instances of a given schema. GML is exactly such a schema intended for collection and transfer of geographic information. The term geographic information should be understood as both geometry and attributes (properties) of geographic features. GML specification defines mechanisms and syntax GML uses for geographic information recording in XML. The original idea behind GML creation was application of XML for information recording for its lossless interchange, independent of hardware and software platform, and independent of character and technology of the information system.

## 2. DATA

To check the ability of GIS software to read GML files, the authors chose available commercial system ArcGIS and two free, open source programs: Open Jump and Quantum GIS. Analyses were based on data taken from Topographical Database (TBD), recorded in GML in accordance with specification published as technical guidelines by Surveyor General of Poland (Główny Geodeta Kraju 2003). These data were recorded in several dozen files, each containing information about one thematic layer (buildings, roads, rivers and so on). Additionally, publicly accessible schema files, describing the required structure of these GML files were downloaded from Surveyor's General web page. The structures of GML file and its corresponding schema file are shown below (fig.1, fig. 2).

```
<?xml version="1.0" encoding="utf-8"?>
<Plik_TBD xmlns="http://www.gugik.gov.pl/TBD"
 xmlns:gml="http://www.opengis.net/gml"
 xmlns:xlink="http://www.w3.org/1999/xlink"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.gugik.gov.pl/TBD TBDGML_ver1_34.xsd">
    <Metadane>
    <Dane>
        <ADGM_A>
            <X_KOD_TBD>ADPA01</X_KOD_TBD>
            <X_AKTUALNOSC_G>2006-02-28</X_AKTUALNOSC_G>
            <X_AKTUALNOSC_A>2006-02-28</X_AKTUALNOSC_A>
            <X_KAT_DOKL_GEOM>2</X_KAT_DOKL_GEOM>
            <X_ZRODLO_DANYCH_G>PRG</X_ZRODLO_DANYCH_G>
            <X_ZRODLO_DANYCH_A>PRG</X_ZRODLO_DANYCH_A>
            <X_KAT_ISTNIENIA>1</X_KAT_ISTNIENIA>
            <X_RODZAJ_REPR_GEOM>ZU</X_RODZAJ_REPR_GEOM>
            <X_DATA_UTWORZENIA>2006-11-21</X_DATA_UTWORZENIA>
            <X_DATA_MODYFIKACJI>2006-03-07</X_DATA_MODYFIKACJI>
            <ID>1</ID>
            <ID_TERYT>3064011</ID_TERYT>
            <NAZWA>M. Poznan</NAZWA>
            <ID_POWIATU>3064</ID_POWIATU>
            <OBSZAR>
                <gml:Polygon>
                    <gml:outerBoundaryIs>
                        <gml:LinearRing>
                            <gml:coord>
                                <gml:X>364025.18</gml:X>
                                <gml:Y>507520.82</gml:Y>
                            </gml:coord>
                            ...
                            <gml:coord>
                                <gml:X>375466.77</gml:X>
                                <gml:Y>508450.76</gml:Y>
                            </gml:coord>
                        </gml:LinearRing>
                    </gml:outerBoundaryIs>
                </gml:Polygon>
            </OBSZAR>
        </ADGM_A>
    </Dane>
</Plik_TBD>
```

**Fig. 1**. The structure of GML file.

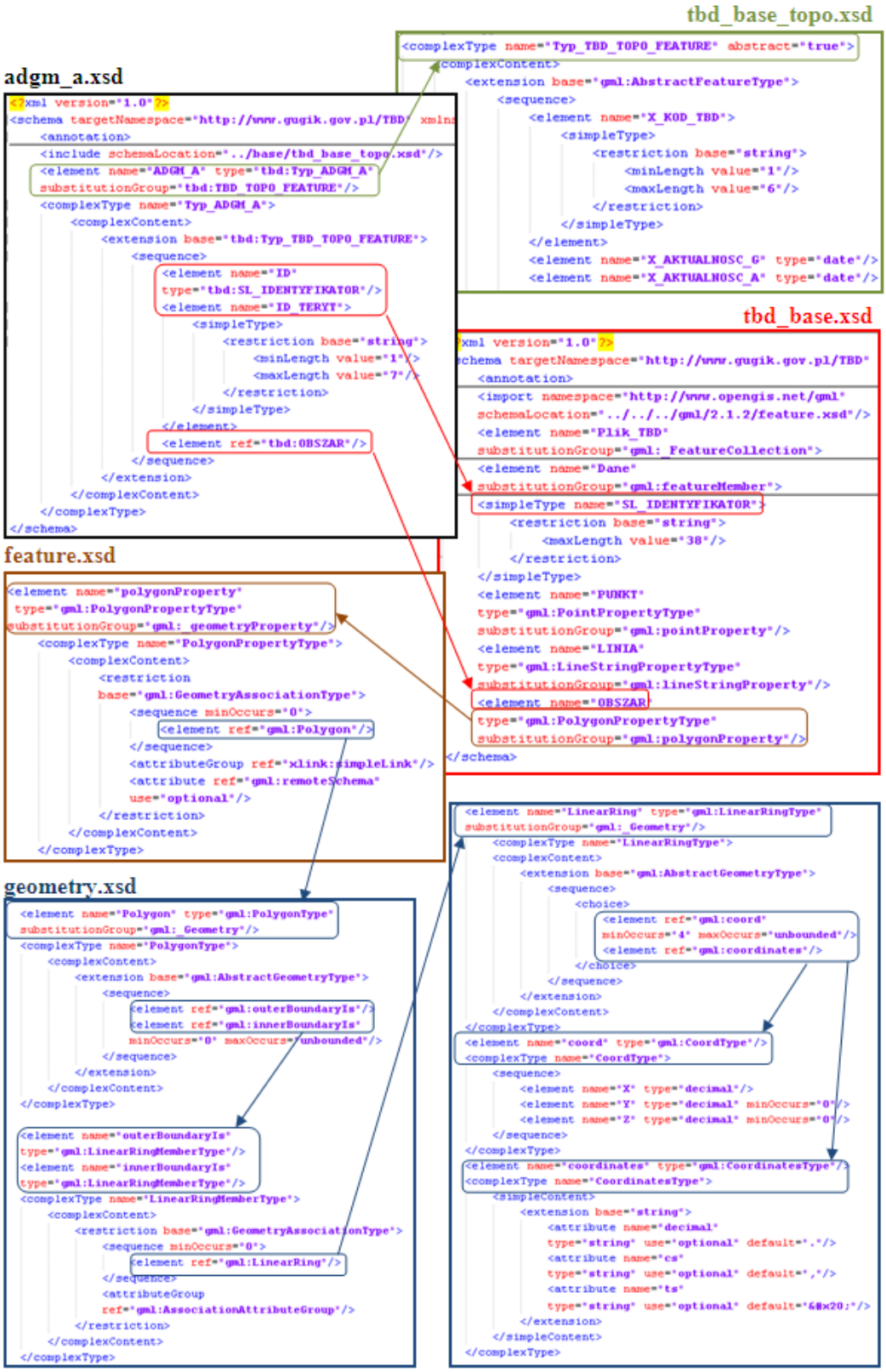


**Fig. 2**. Definition of the ADGM_A object attributes.

Presented above part of the GML file shows an ADGM_A object from the TBD database. First line contains information about the XML version and the character encoding used. In the next tag, there is information about location of TBD files and their schema files, which are essential, because they define objects in TBD files. Next lines include ADGM_A metadata (<Metadane>) and ADGM_A data (<Dane>). The Dane tag consists of descriptive and geometric data (<OBSZAR>).

Each GML file should have its corresponding schema file, because only then programs can read this GML file correctly. In case of TBD, it is needed to add the whole set of schema files. The example presented below, shows exact definition of three attributes (<ID>, <X_KOD_TBD> and <OBSZAR>) of the AGDM_A object (fig. 2).

This example presents only part of schema files that define these three selected attributes. The first of presented schema files (adgm_a.xsd) is the main schema file directly connected to GML file containing ADGM_A object., There are references in this file to other schema files, which contain more and more precise definition of each attribute of ADGM_A object.

After analyses of all these schema files, it can be concluded that the <ID> attribute is a string type, limited to 30 characters. The <X_KOD_TBD> attribute is a string type as well, and it can be one to six characters long. The <OBSZAR> attribute is a sequence of coordinates, which locate this object in a geographical space. As it can be seen, the structure of mutually related schema files is rather complicated. To correctly load spatial data recorded in GML, the user has to possess at least basic knowledge in the field of GML and has to be familiar with underlying TBD specification.

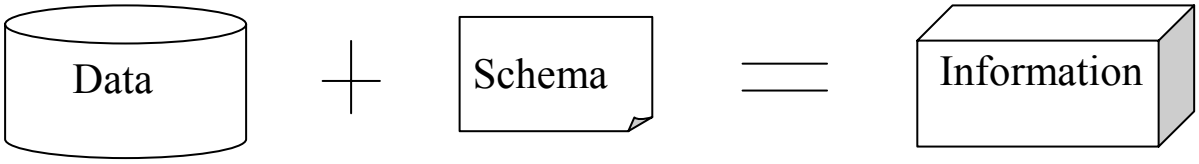


**Fig. 3.** Relation between GML file and its corresponding schema file.

As it was mentioned before, the structure of each GML file is defined by schema file. This schema file is necessary to get full information about objects contained in GML file (fig. 3). Authors' experiences show, that this division into two separate elements is the major cause of problems regarding handling GML files by GIS software. Results of undertaken attempts are presented below.

## 3. SOFTWARE

**ArcGIS 9.3** – this program contains special extension – Data Interoperability, which allows the user to convert files from one format to another and define his own input and output formats. In case of GML files, it also allows to choose suitable schema files. The list of executed operations and their results presented on the screen is very helpful as well, because it allows to pick up mistakes, especially those caused by incorrect location of GML files in relation to schema files. Aforesaid example shows, that the main schema file contains a

number of associated schema files complementing it. Program has to locate this set of schema files to load spatial data correctly. It is possible, because each of schema file includes relative access paths to other schema files associated with it.

In practice, the program managed to load one spatial object from one GML file. In case of several dozen GML files, it was necessary to point out this set of GML files and one, overriding schema file (fig. 4).

```
<?xml version="1.0"?>
<schema targetNamespace="http://www.gugik.gov.pl/TBD"
    <annotation>
    <include schemaLocation="features/adgm_a.xsd"/>
    <include schemaLocation="features/adms_a.xsd"/>
    <include schemaLocation="features/adoe_a.xsd"/>
    <include schemaLocation="features/adol_a.xsd"/>
    <include schemaLocation="features/adpd_a.xsd"/>
    <include schemaLocation="features/arad_p.xsd"/>
    <include schemaLocation="features/bbbd_a.xsd"/>
    <include schemaLocation="features/bbcm_a.xsd"/>
    <include schemaLocation="features/bbhy_a.xsd"/>
    <include schemaLocation="features/bbhy_l.xsd"/>
    <include schemaLocation="features/bbib_a.xsd"/>
    <include schemaLocation="features/bbiu_a.xsd"/>
    <include schemaLocation="features/bbiu_p.xsd"/>
```

**Fig. 4.** Fragment of tbd_topo.xsd schema file including relative access paths to schema files of individual TBD objects.

**OpenJUMP** – one of the most interesting pieces of free software in the field of GML files import. This program allows the user to load spatial data compliant with its own schema file (JUMP GML – *.jml) and FME GML files (*.gml, *.xml, *.fme). But additionally it gives the opportunity to use schema file (or more precisely – template file, because it has structure completely different from schema file). The structure of this template (fig. 5) is specified in JUMP Workbench User's Guide (Ministry Of Sustainable Resource Management 2003).

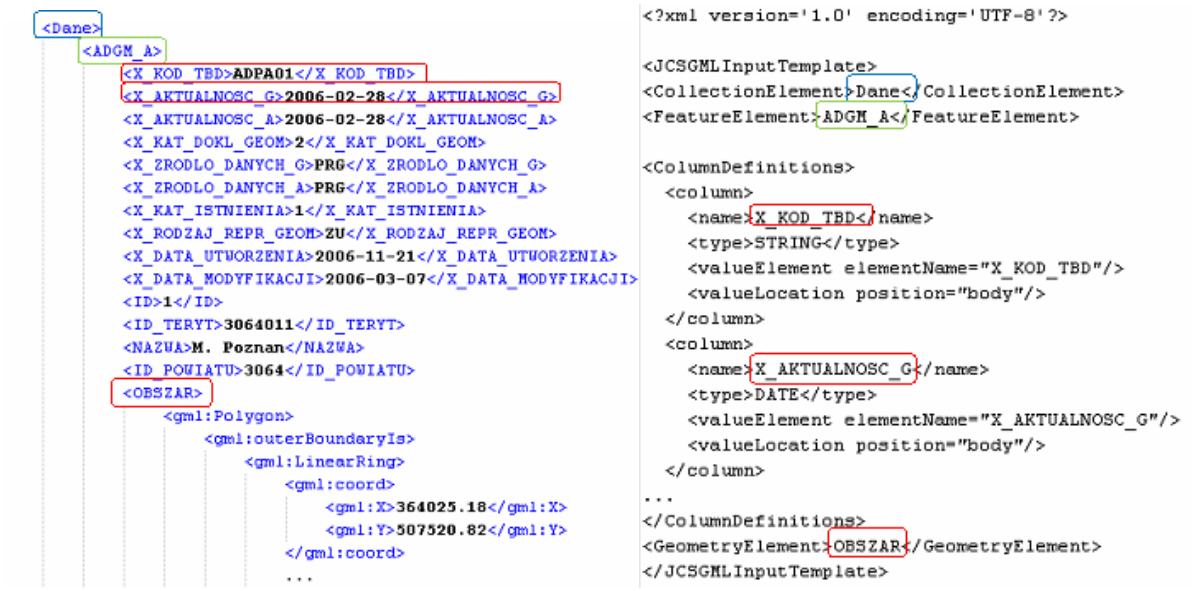


**Fig. 5.** Fragment of GML file with ADGM_A object (on the left) and fragment of template file defining attributes of ADGM_A object (on the right).

Creation of this template file is not easy, because it requires deep knowledge of the data recorded in GML file. Another important thing is that this software unfortunately cannot load data which does not contain geometric information (tables with descriptive information about objects only)

**Quantum GIS** – this program uses The OGR Simple Features Library (GDAL 2009) to import GML files. It is a C++ open source library (and command line tools) providing read (and sometimes write) access to a variety of vector file formats including ESRI Shapefiles, PostGIS and Mapinfo formats.

In contrast to most GML readers, the OGR GML reader makes no effort to read the format XML Schema definition of the feature classes in a GML file. Instead it attempts to automatically discover them and their associated properties by scanning the file and looking for "known" GML objects in the gml namespace to determine their organization. Unfortunately Quantum GIS failed to load TBD data. The program only displayed the error message stating that opened file "is not a valid or recognized data source".

Mixed result of TBD files import led the authors to try loading another files. Since TBD is the only formal GML usage in Poland as yet, they used, available on the internet, exemplary files coming from AAA-NAS (Germany), Ordnance Survey MasterMap (Great Britain) and TOP10NL (Netherlands).

**ArcGIS** finally managed to load them even without running Data Interoperablity extension, but it can be supposed that it used, recorded in this extension, profiles for the last two loaded data sources (OS Master Map and TOP10NL). The authors are not sure how it loaded AAA-NAS data, but probably the support for this format is (similarly to the two mentioned above) embedded in this extension by its producer.

As for **Quantum GIS** the authors obtained mixed results. OS MasterMap caused least problems. In the first take all objects were loaded and presented on one layer, regardless of their geometry type (point, line, polygon) and set of attributes, which finally caused omission of these attributes. To dissolve this problem, specific feature of OGR Library was used. The first time a GML file is opened it is completely scanned in order to determine the set of geographic feature types, the attributes associated with each and other dataset level information. Even the contents of fields are scanned to try and determine the type of the field. This information is stored in a .gfs file (fig. 6) with the same base name as the target GML file. Subsequent accesses to the same GML file will use the .gfs file to predefine dataset level information accelerating access. This .gfs file was manually edited to alter how the GML file was parsed and only one feature class was left. This allowed loading chosen feature class along with full set of attributes. Similar procedure was applied to AAA-NAS file, but both in case of the whole file and its part, despite proper structure recognition, no features were loaded. And finally attempt to open TOP10NL file ended up with error message stating that opened file "is not a valid or recognized data source".

```
<GMLFeatureClassList>
  <GMLFeatureClass>
    <Name>AX_Fahrwegachse</Name>
    <ElementPath>AX_Fahrwegachse</ElementPath>
    <DatasetSpecificInfo>
      <FeatureCount>1350</FeatureCount>
    </DatasetSpecificInfo>
    <PropertyDefn>
      <Name>breiteDesVerkehrsweges</Name>
      <ElementPath>breiteDesVerkehrsweges</ElementPath>
      <Type>Integer</Type>
    </PropertyDefn>
    <PropertyDefn>
      <Name>funktion</Name>
      <ElementPath>funktion</ElementPath>
      <Type>Integer</Type>
    </PropertyDefn>
    <PropertyDefn>
      <Name>hatDirektUnten</Name>
      <ElementPath>hatDirektUnten</ElementPath>
      <Type>Untyped</Type>
    </PropertyDefn>
  </GMLFeatureClass>
</GMLFeatureClassList>
```

**Fig. 6.** Example of .gfs file.

## 4. CONCLUSION

Although Geography Markup Language formally exists for several years it never gained the appropriate place among Geographic Information Systems users. It is not surprising. Presented above results of analyses conducted by the authors seems to explain this situation. More and more programs advertise themselves as being able to read GML files. But only some of them allow to simultaneously specify the file to read and its corresponding schema.

Other approaches (simplified schema, attempt to guess the file structure) can also give positive results but they require user to have knowledge of at least basics of GML. We should wish ourselves creation of free application allowing loading of GML along with schemas. Because the biggest advantage of GML is possibility to record spatial data depending on data specifics and user needs. The authors hope that development of GIS software will lead into better handling of GML files.

## REFERENCES


Cichociński P. 2001, Język XML i jego implementacje dla danych przestrzennych (XML and its implementations for spatial data), XI Konferencja Naukowo – Techniczna Systemy Informacji Przestrzennej, Warszawa.

ESRI 2008, ArcGIS 9.3 Desktop Help.

Główny Geodeta Kraju 2003, Wytyczne techniczne – Baza Danych Topograficznych (Technical Guidelines – Topographical Database), Główny Urząd Geodezji i Kartografii, Warszawa.

ISO 19136:2007 Geographic information – Geography Markup Language (GML).

Ministry Of Sustainable Resource Management 2003, JUMP Workbench User's Guide, www.vividsolutions.com/JUMP/bin/JUMP%20User%20Guide.pdf.

GDAL 2009, OGR Simple Feature Library, http://www.gdal.org/ogr/index.html.


## CONTACTS


Ms. Izabela Śliż
AGH University of Science and Technology
Department of Geomatics
Al. Mickiewicza 30
30-059 Kraków
POLAND
Tel. + 48 12 617 23 00
Fax + 48 12 617 45 88
Email: isliz@agh.edu.pl

Dr. Piotr Cichociński
AGH University of Science and Technology
Department of Geomatics
Al. Mickiewicza 30
30-059 Kraków
POLAND
Tel. +48 12 617 34 31
Fax + 48 12 617 45 88
Email: Piotr.Cichocinski@agh.edu.pl