

INTEGRACION DE SISTEMAS DE INFORMACION GEOGRAFICA

Jacqueline Guzman, Regina Motz

jguzman@adinet.com.uy, rmotz@fing.edu.uy

Instituto de Computación - Facultad de Ingeniería
Universidad de la República - Uruguay

RESUMEN

Este trabajo trata la integración de información geográfica proveniente de fuentes heterogéneas. Se presentan algunos estándares y formatos de datos utilizados en los SIG's (Sistemas de Información Geográfica), se analiza la integración de la información proveniente de estas fuentes, y se plantea la definición de un esquema de intercambio basado en XML junto con las transformaciones necesarias para llevar uno de los estándares a dicho esquema. En particular en este trabajo se establecen las correspondencias entre los conceptos geográficos del formato de datos para información geográfica de ESRI (Shapefile) y el modelo GML V2.0 de OpenGis, y se especifican las transformaciones necesarias para llevar información de uno a otro.

1. INTRODUCCION

En los últimos años las empresas y organizaciones autónomas han tomando conciencia de la importancia que tiene compartir su información, o parte de la misma, para tener un mejor y más completo conocimiento de la realidad a menor costo, reaprovechando los datos existentes y extendiendo las funcionalidades de sus sistemas. Sumado a esto surgen nuevas tecnologías como los Sistemas de Información Geográfica (SIG), que permiten presentar los datos en forma consolidada y ubicados geográficamente. Es en este marco que se presenta la necesidad de estudiar las formas en que se puede integrar la información geográfica proveniente de distintas fuentes, sin olvidar que hoy día la herramienta más conocida para ofrecer y obtener información es Internet.

Es entonces en este escenario que el proyecto realiza el análisis de las técnicas de extracción e integración de información geográfica provenientes de sistemas heterogéneos, con especial énfasis en el estudio de los estándares.

Cada SIG tiene su forma de generar y almacenar los datos, los cuales pueden encontrarse en: base de datos relacionales, páginas html, etc., y pueden seguir o no un estándar. A la hora de compartir datos las empresas tienen que establecer los formatos en los cuales brindarán la información (o esperará recibirla), y el que la recibe deberá convertirla al formato utilizado por su sistema, es aquí donde los costos y la automatización de dicha tarea se beneficiará si los formatos utilizados siguen algún estándar. Por esta razón existen varias propuestas de estándares para modelar e intercambiar datos geográficos, como por ejemplo los propuestos por: el consorcio OpenGIS, los de Spatial Data Transfer Estándar (SDTS), y otras propuestas como la integración semántica guiada por ontologías para asegurar la integración semántica en el ámbito conceptual. En este trabajo se presenta el estudio de algunos de ellos, específicamente: GML de OpenGis, Shapefile de ESRI, "Content Standards for Digital Geospatial Metadata" del FGDC (Federal Geographic Data Comité), y otros (por más detalles del relevamiento de los estándares se puede consultar [7])

Resumiendo: este trabajo presenta algunos estándares y formatos de datos utilizados en los SIG's, analiza la integración de la información proveniente de estas fuentes, y plantea la definición de un esquema de intercambio basado en XML junto con las transformaciones necesarias para llevar uno de los estándares a dicho esquema. En particular, este trabajo establece correspondencias entre los datos y los conceptos geográficos, y muestra las transformaciones necesarias para llevar información en formato Shapefile a GML y viceversa.

El artículo se organiza de la siguiente forma. En la Sección 2 se da una descripción general de la solución propuesta para la integración de datos geográficos. Se describen brevemente los problemas encontrados y se ubica el problema

de transformación que desarrolla este trabajo. En la Sección 3 se especifican las transformaciones entre GML y Shapefile. La Sección 4 presenta algunas conclusiones y trabajos futuros.

2. SOLUCION GENERAL

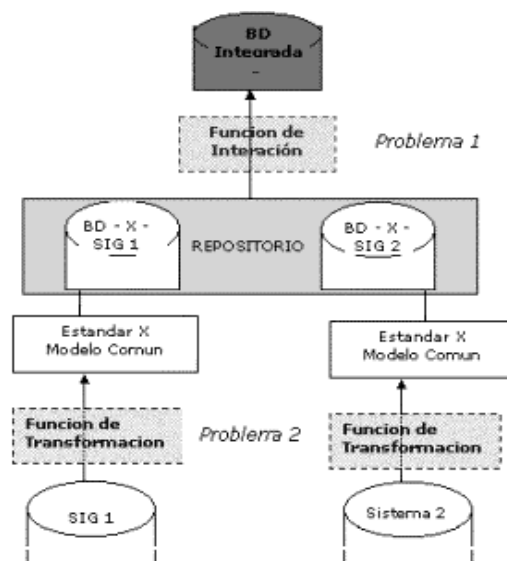
El problema original planteado parte del objetivo de integrar datos geográficos que se encuentran en repositorios heterogéneo, pudiéndose comunicar dos sistemas cualesquiera, o utilizar un Centro de Datos y cada sistema integrar los datos que desee con dicho centro. Este último es la forma de trabajo elegida debido a que si el intercambio es:

1. Entre dos SIGS cualesquiera, puede darse una de las siguientes situaciones:
 - las dos fuentes de datos siguen el mismo estándar.
 - las dos fuentes de datos siguen distintos estándares. Habrá que analizar ambos y realizar una transformación de esquemas y llevar los datos a uno de los dos esquemas, esta transformación (semántica y sintáctica) podrá ser reutilizada para otros sistemas que sigan estos estándares
 - Una de las fuentes sigue un estándar mientras que la otra fuente de datos no sigue ningún estándar. Habrá que realizar la transformación sintáctica y semántica del modelo de datos propietario al estándar y viceversa. Estas transformaciones no podrán ser reutilizadas por el uso de formatos propietarios.
 - ninguna de las fuentes de datos siguen estándar alguno. Habrá que realizar las transformaciones semánticas y sintácticas entre ambos modelos de datos, las cuales no podrán ser reutilizada.

Para cada sistema se deberá implementar las transformaciones de esquema de su estándar a todos los formatos manejados por cada uno de los sistemas que quieran compartir sus datos, y de cada uno de ellos al nuevo sistema, en el caso que ya no estén implementadas. Se tendrá una matriz de transformación de esquemas entre estándares (nxn), y en caso de formatos propietarios habrá que implementar los casos particulares.

2. Con un Centro de Datos y todo SIG realizar el intercambio con dicho centro. Esta es una simplificación del caso antes descrito en el cual se reducen la cantidad de transformaciones (matriz) a la mitad. Se elige el estándar que manejará el Centro de Datos y para cada sistema se implementará las transformaciones necesarias que permitan llevar su modelo de datos al modelo canónico (esquema del centro de datos según estándar elegido) [1] y viceversa.

Considerando la simplificación al problema expuesta anteriormente, una de las preguntas que ha quedado sin responder es: ¿ cuáles son los problemas de integración a resolver ? El problema se ha simplificado quedando la situación que ilustra la siguiente figura:



Problema 1: Es la Integración de las distintas bases de datos, las cuales respetan el modelo común y están parcialmente llenas según los datos que se pudieron exportar respetando el estándar de dicho modelo.

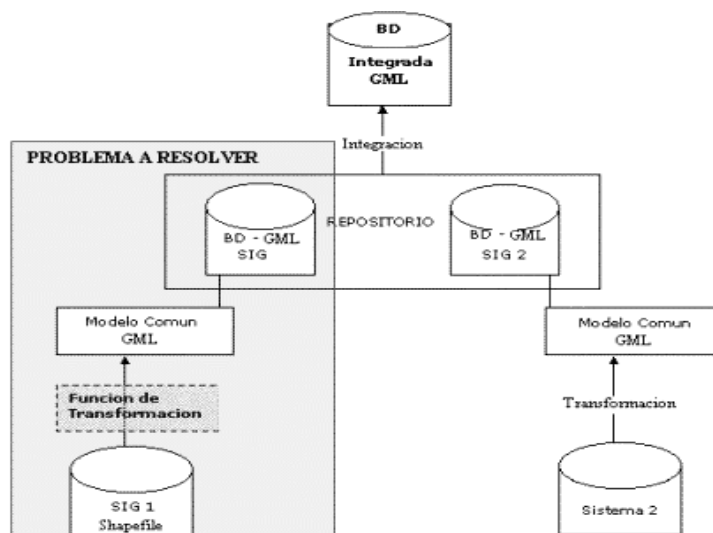
Problema 2: Transformar los datos que están en el SIG, o sistema real, al modelo común elegido. Algunos datos no se podrán exportar y puede que falten otros datos para completar totalmente los datos esperados en el modelo.

En este trabajo nos concentraremos en la resolución del segundo problema: transformar los datos de los distintos formatos utilizados por los clientes, al estándar /formato del centro de datos.

Para poder cumplir el objetivo planteado debemos:

1. Elegir un estándar para el centro de datos y definir el modelo de datos común según dicho estándar.
Considerando la solvencia de los estándares definidos por el Consorcio OpenGis y su trabajo en el área de SIGs, elegiremos utilizar como modelo estándar del centro de datos el definido por esta comunidad: GML.
Si bien la arquitectura propuesta presenta XML para el modelo común por ser la tecnología más apropiada para el transporte de información vía Internet, vale observar que GML es un modelo conceptual para información geográfica escrito en XML.
En particular se utilizará GML versión 2.0 en la cual los tipos de datos no se definen mediante DTD's, como en la versión 1.0, sino que define esquemas XML para tales efectos.
2. Para todos los posibles formatos que puede tener los modelos de datos de las agencias o clientes, se deberá definir las transformaciones que necesarias para que integren la comunidad, es decir que puedan aportar y/o extraer información con el centro de datos. La fuente de datos puede:
 - a) seguir el estándar elegido para el modelo canónico. hay que integrar los datos problema
 - b) no estar estructurada, por ejemplo la información que se pueda extraer de páginas web
 - c) ser una base de datos. En este caso pueden ser datos estructurados que no siguen el estándar del modelo común. Se requerirá entonces especificar las transformaciones entre los esquemas del centro de datos y el de la fuente de datos.

Se elegirá estudiar las transformaciones necesarias para algún ejemplo representativo. Considerando la difusión del formato definido por ESRI, shapefile, y que este es abierto, se tomara este formato como caso de estudio. La siguiente figura ubica en el escenario general de integración planteado el problema que desarrolla este trabajo.



3. TRANSFORMACIONES ENTRE GML Y SHAPEFILE

3.1 FORMATOS

Antes de mostrar las transformaciones necesarias para realizar correspondencias entre los formatos GML y Shapefile se describe a continuación muy brevemente las características de cada uno de ellos. Información más detallada puede encontrarse en [3,4] y [2] para GML y Shapefile respectivamente.

CARACTERISTICAS PRINCIPALES DE GML V 2.0

- Es XML para información geográfica por lo que es ideal para cargar y/o transportar información independientemente de la plataforma.
- Es un Modelo de datos que ofrece un conjunto de elementos *geográficos* básicos: punto, línea, polígonos, etc., con información asociada como: sistema de referencia, identificación, etc.
- Es un Lenguaje por lo que ofrece herramientas que permiten definir otras geometrías según necesidades.
- Versión 1.0 - DTD's & Versión 2.0 Esquemas
- Maneja elementos 3D y no modela topología.

CARACTERISTICAS PRINCIPALES DEL FORMATO SHAPEFILE

- Es un estándar de *geometrías*, especifica: puntos, líneas, polígonos, etc. No especifica información como: sistema de referencia, escalas, etc., necesaria para interpretar los datos como información geográfica.
- Está formado por tres archivos: .shp, .dbf, .shx (Shp y Shx: son usados para especificar las geometrías).
- Dbf es usado para asociar información descriptiva a la geométrica. No especifica qué ni cómo será modelada.
- Cada shp contiene elementos de igual geometría.
- Maneja elementos en 3D y no modela topología.

3.2 TRANSFORMACIONES

Para realizar las transformaciones entre GML y SHAPEFILE y viceversa, consideramos las siguientes presupuestas:

DE SHAPEFILE A GML

Para que la transformación sea útil partimos del presupuesto que ya existe un protocolo en el que se acuerdan y brindan los datos para interpretar la información, como: sistemas de referencia, escalas, etc.

La idea general es recorrer el archivo Shp y para cada elemento que lo compone, extraer y transformar la información geométrica.

DE GML A SHAPEFILE

El conjunto de geometrías válidas para shp son fijas, por lo cual solo se transformarán aquellos elementos Gml que tengan un correspondiente dentro del dominio Shp.

En este trabajo mencionamos pero no especificamos las transformaciones de los tipos propietarios.

No se especificarán las transformaciones de las geometrías que sintácticamente sean iguales a otro pero difieran en cosas como el rol.

3.3 TRANSFORMACIONES DE PUNTOS

En SHP: Points es un par de coordenadas con la siguiente representación:

```
Point
{ Double X // coordenada X
  Double Y // coordenada Y }
```

En GML: Point es una coordenada o tupla con la siguiente representación:

```
<complexType name="PointType">
  <complexContent>
    <extension base="gml:AbstractGeometryType">
      <sequence>
        <choice>
          <element ref="gml:coord"/>
          <element ref="gml:coordinates"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Podemos observar que existe **igualdad semántica** en concepto y representación. La transformación será entonces entre las representaciones. Las transformaciones especificadas son las siguientes: F1, F2 y F3 como se detalla a continuación:

- F1 = Transf_PuntoShpToGml: {PointSHP} à {PointGML}
- F2 = Transf_PuntoZShpToGml: {PointZSHP} à {PointGML}
- F3 = Transf_PuntoGmlToShp: {PointGML} à {PointSHP}U{PointZSHP}

Observamos que estas transformaciones poseen las propiedades: de ser: biyectivas, totales y además se cumple que:

$$F1 \cup F2 = F3^{-1} ; F1^{-1} = F3 ; F2^{-1} = F3.$$

Por lo tanto son: injectivas y subjectivas.

De acuerdo a las propiedades que encontramos en las transformaciones podemos derivar las conclusiones siguientes:

- No existe pérdida de información (en cuanto a la geometría se refiere)
- Es posible obtener la geometría original a partir del elemento transformado.
- Es posible transformar puntos en 3D de Gml a Shp

3.4 TRANSFORMACIONES DE CONJUNTO DE PUNTOS

En SHP: Conjunto de puntos, siendo Point un par de coordenadas: x, y se representa por:

```
MultiPoint
{ Double[4]      Box // Bounding Box: Xmin, Ymin, Xmax, max
  Integer        NumPoints // Números de puntos
  Point[NumPoint] Points // Conjunto de puntos }
```

En GML: Conjunto de puntos se representa por:

```
<complexType name="MultiPointType">
  <complexContent>
    <restriction base="gml:GeometryCollectionType">
      <sequence>
        <element name="pointMember" maxOccurs="unbounded">
          <complexType>
            <sequence> <element ref="gml:Point"/> </sequence>
          </complexType>
        </element>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

Se observa que la representación shapefile agrega el concepto de Box, por lo cual el concepto es igual entre ambos estándares, pero la representación NO. Para no perder información definimos una nueva geometría en GML que representa un MultiPoint considerando el Box. (Recordar que Gml tiene una geometría Box igual a la de shp). La transformación en GML quedaría entonces:

```
<complexType name="N_PtosConMarcoType">
  <complexContent>
    <extension base="gml:AbstractGeometryType">
      <element ref="gml:Box">
        <element ref="gml:MultiPoint">
      </extension>
    </complexContent>
  </complexType>
```

De esta forma tenemos la transformación de Shp a Gml (F1) siguiente:

$$F1 = \text{Transf_MultiPointShpToGml}: \{\text{MultiPointSHP}\} \rightarrow \{\text{N_PuntosConMarcoGMLExt}\}$$

Donde podemos observar que F1 es total (definida para todo elemento del dominio) y biyectiva (fácilmente se podría definir la función inversa).

Todo lo anterior es válido para elementos MultiPointZ de Shp, mediante la transformación:

$$F2 = \text{Transf_MultiPuntoZShpToGml}: \{\text{MultiPointZSHP}\} \rightarrow \{\text{N_LineConMarcoGML}\}$$

En las transformaciones de Gml a Shp se cumple que:

- La geometría MultiPoint Gml deberá ser llevada a MultiPoint-Z de Shp.
- Estas transformaciones deberán calcular el box que contiene a la geometría.

La transformación es:

$$F3 = \text{Transf_MultiPuntoGmlToShp}: \{\text{MultiPointGML}\} \rightarrow \{\text{MultiPointSHP}\} \cup \{\text{MultiPointZSHP}\}$$

donde se puede observar que F3 es total, todo elemento del dominio puede ser transformado, además F3 no es biyectiva. La función inversa devuelve un objeto válido en Gml pero no el original. Es posible transformar MultiPoint en 3D de Gml a Shp.

3.5 TRANSFORMACIONES DE LINEAS

En SHP: PolyLine es conjunto ordenado de vértices que forman particiones (segmentos de línea), que pueden o no estar conectadas o interceptarse. Su representación es la siguiente:

PolyLine

```
{ Double[4]      Box    // Bounding Box: Xmin, Ymin, Xmax, max
  Integer        NumParts // Números de partes en la polilínea
  Integer        NumPoints // Total de números de puntos
  Integer [NumParts] Parts // Array de nro de partes, c/lugar=índice al 1º pto de esa parte, en el array de ptos
  Point[NumPoint] Points // Conjunto de puntos. Ptos de la 1º parte, luego los ptos de la 2º parte, ...) }
```

En GML: MultiLineString es una colección de segmentos de línea. Su representación es la siguiente:

```
<complexType name="MultiLineStringType">
  <complexContent>
    <restriction base="gml:GeometryCollectionType">
      <sequence>
        <element name="lineStringMember" maxOccurs="unbounded">
          <complexType>
            <sequence>
              <element ref="gml:LineString"/>
            </sequence>
          </complexType>
        </element>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

Observamos que ambas representaciones son conceptualmente equivalentes pero Shapefile agrega un Box que Gml no. Para no perder información se define una nueva geometría en GML que representa un PolyLine Shp considerando el Box

```
<complexType name="N_LineasConMarcoType">
  <complexContent>
    <extension base="gml:AbstractGeometryType">
      <element ref="gml:Box">
      <element ref="gml:MultiLineString">
    </extension>
  </complexContent>
</complexType>
```

En las transformaciones de Shp a Gml la transformación es:

$$F1 = \text{Transf_PolyLineShpToGml: } \{ \text{PolyLineSHP} \} \rightarrow \{ \text{N_LineConMarcoGML} \}$$

donde F1 es total (definida para todo elemento del dominio) y es biyectiva (fácilmente se podría definir la función inversa).

Todo lo anterior es también válido para elementos PolyLineZ de Shp, mediante la transformación

$$F2 = \text{Transf_PolyLineZShpToGml: } \{ \text{PolyLineZSHP} \} \rightarrow \{ \text{N_LineConMarcoGML} \}$$

En las transformaciones de Gml a Shp:

- Las geometrías GML que modelan líneas son: LineString (segmento de línea) y MultiLineString (conjunto de LineString).
- Las geometrías SHP que representan líneas son: PolyLine y PolyLineZ.
- Estas transformaciones calcularán el box que contiene a la geometría.

Las transformaciones son:

F3 = Transf_LineStringGmlToShp: {LineStringGML} à {PolyLineSHP} U {PolyLineZSHP}

F4 = Transf_MultiLineStringGmlToShp: {MultiLineStringGML} à {PolyLineSHP} U {PolyLineZSHP}

Se observa que F3 y F4 son totales (todo elemento del dominio puede ser transf.) y no son biyectivas. La función inversa Transf_PolyLineShpToGml U Transf_PolyLineZShpToGml, devuelven un objeto válido en Gml pero no el original. Es posible transformar líneas en 3D de Gml a Shp.

3.6 TRANSFORMACIONES DE POLIGONOS

En SHP: Polygon es conjunto de anillos que no se interceptan ni se tocan en los vértices, puede contener muchos anillos exteriores. Un anillo es una secuencia ordenada de vértices, tal que en sentido horario determinan polígonos simples y en sentido antihorario determinan polígonos hoyos.

Polygon

{ Double[4] Box // Bounding Box: Xmin, Ymin, Xmax, max

Integer NumParts // Números de partes de anillos del poligono

Integer NumPoints // Total de números de puntos

Integer [NumParts] Parts // Array de nro de partes, c/lugar=índice al 1º pto de esa parte, en el array de ptos.

Point[NumPoint] Points // Conjunto de puntos. Ptos de la 1º parte, luego los ptos de la 2º parte, ...}

Cada parte representa un anillo orientado sin embargo el dato de qué anillos son los exteriores no se dispone.

En GML: Polygon es una superficie conectada. El borde de un polígono es un conjunto de anillos (LinearRing). Se distingue el borde exterior de los interiores y los anillos interiores no pueden atravesarse ni tocarse entre si.

```
<complexType name="PolygonType">
  <complexContent>
    <extension base="gml:AbstractGeometryType">
      <sequence>
        <element name="outerBoundaryIs">
          <complexType>
            <sequence>
              <element ref="gml:LinearRing"/>
            </sequence>
          </complexType>
        </element>
        <element name="innerBoundaryIs" minOccurs="0" maxOccurs="unbounded">
          <complexType>
            <sequence>
              <element ref="gml:LinearRing"/>
            </sequence>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

En las transformaciones de Shp a Gml se debe considerar:

- La representación Shp no diferencia anillos exteriores e interiores, la transformación de shp a gml deberá calcularlo.
- Para poder reconstruir el polígono original la transformación debiera respetar el orden de los puntos (en Shp los anillos son orientados)
- Los polígonos shp son 'limpios': sus anillos cumplen la restricción exigida por Gml de que los anillos interiores no deben cortarse ni atravesarse.
- Para no perder la información 'box', se define la siguiente geometría Gml :


```

<complexType name="N_PolygonConMarcoType">
  <complexContent>
    <extension base="gml:AbstractGeometryType">
      <element ref="gml: Box">
      <element ref="gml: Polygon">
    </extension>
  </complexContent>
</complexType>

```

La transformación es:

$$F1 = \text{Transf_PolygonShpToGml}: \{ \text{PolygonSHP} \} \rightarrow \{ \text{N_PolygonConMarcoGML} \}$$

Se observa que F1 es total y biyectiva, se podría definir la función inversa.

Todo lo anterior es válido para elementos PolygonZ de shapefile, mediante la transformación:

$$F2 = \text{Transf_PolygonZShpToGml}: \{ \text{PolygonZSHP} \} \rightarrow \{ \text{N_PolygonConMarcoGML} \}$$

En las transformaciones de Gml a Shp:

- La geometría Polygon Gml debiera ser llevada a Polygon-Z de Shp.
- Estas transformaciones deberán calcular el Box que contiene a la geometría.
- Los polígonos shapefile deben ser limpios, los anillos interiores del polígono Gml cumple esto, solo se deberá verificar que el anillo exterior lo cumpla.
- En ambos modelos los anillos son cerrados.
- Debe tenerse cuidado con el orden en que se procesen los vértices, el sentido horario especifica como zona del polígono el interior del anillo.

La transformación es:

$$F3 = \text{Transf_PolygonGmlToShp}: \{ \text{PolygonGML} \} \rightarrow \{ \text{PolygonSHP} \} \cup \{ \text{PolygonZSHP} \}$$

F3 no es total. Si el anillo exterior del polígono tiene un lado adyacente a alguno de los anillos interiores no puede ser transformado. F3 tampoco es biyectiva, las funciones inversas Transf_PolygonShpToGml o Transf_PolygonZShpToGml, devuelven un objeto válido en Gml pero no el original.

3.7 TRANSFORMACIONES DE CONJUNTO DE POLIGONOS Y GEOMETRIAS

No existe geometría shp que modele un conjunto de polígonos.

La transformación será:

$F: \{ \text{MultiPolygonGml} \} \rightarrow \{ \text{PolygonShp} \}$ donde F no es total (no todo polígono Gml es válido para Shp) y tampoco es biyectiva (la inversa no devuelve el objeto original, por lo cual se pierde información).

$$F^{-1}(\{ \text{PolygonShp} \}) = \{ \text{PolygonGml} \}$$

La Multygeometry Gml esta en iguales condiciones que los MultyPolygon.

4 CONCLUSIONES

En este trabajo se establecieron correspondencias entre los conceptos geográficos del formato de datos para información geográfica de ESRI (Shapefile) y el modelo GML V2.0 de OpenGis, y se especificaron las transformaciones necesarias para llevar información de uno a otro.

De estas transformaciones se puede observar que todo elemento Shapfile se puede transformar a una geometría Gml sin pérdida de información y que a su vez todo elemento Shapefile que se lleve a GML puede recuperarse con la función correspondiente de Gml a Shp. Sin embargo la transformación GML a SHP no es completa pues no todo polígono GML puede transformarse a formato Shapefile, no toda multi geometría tiene un elemento correspondiente, válido en el dominio shapefile y no todas las transformaciones de GML&SHP son biyectivas en el sentido de poder recuperar el objeto original aplicando la transformación correspondiente.

Los MultiPolygon de GML llegan como elementos independientes a SHP por lo que si se aplica la transformación para llevarlos a GML se obtiene un conjunto de N_PolygonM y no el MultiPolygon original. Igual ocurre con la geometría MultyGeometry de GML, con el agravante de que es posible que se pierdan en la transformación algunas de las geometrías que lo componen.

Como trabajo futuro es nuestra intención implementar las transformaciones especificadas y solucionar la integración de datos en el mismo estándar.

REFERENCIAS

- [1].- Artículo: "Implementation Architecture for a National Data Center "
Autores: Steve Ramroop, Richard Pascoe
Second International Conference, INTEROP'99 (Zurich, Switzerland, 1999)
Libro: Interoperating Geographic Information Systems (Lecture Notes in
Computer Science 1580)
- [2].- OGC Geography Markup Language 2.0 -
Opengis Implementatios Specification 20 Februray 2001.
<http://www.opengis.net/gml/01-029/GML2.html>
- [3].- ESRI Shapefile Technical Description - An ESRI white paper July 1998
www.esri.com/library/whitepapers/pdfs/shapefiles.pdf
- [4].- ESRI www.esri.com
- [5].- <http://www.opengis.org/info/brochure/OGC-brochure.htm>
- [6].- Evoluon of EO/GIS Interoperability towards an Integrated Application Infraestructure.
Autor: Gunther Landgraf
Second International Conference, INTEROP'99 (Zurich, Switzerland, 1999)
Libro: Interoperating Geographic Information Systems (Lecture Notes in
Computer Science 1580)
- [7] – Integración de Información Geográfica.
Autor: Jacqueline Guzman. Proyecto de grado de la carrera de Ingeniero en Computación
de la Facultad de Ingeniería, Universidad de la República – Uruguay, 2002.