Visualizzatore ed analizzatore di dati GML su client Web

Fortunati L., Fresta G., Piazza Bonati L.

Ottobre 2002

Istituto di Scienza e Tecnologie dell'Informazione

1. Introduzione

L'OpenGIS Consortium da tempo sta sviluppando il Geography Markup Language (GML), un linguaggio non proprietario specificatamente definito per memorizzare e trasferire (anche via Internet) dati spaziali. Basato sullo standard XML, gestisce sia la geometria che le proprietà degli elementi geografici. Ciò consente alle varie organizzazioni di mettere in condivisione data-set eterogenei ed all'utente di accederli in modo completamente trasparente. Dal punto di vista applicativo, consente di semplificare e standardizzare le operazioni in vari settori: dalla costruzione di mappe alla trasformazione dei dati, dalla interrogazione spaziale all'analisi geografica, incluse le emergenti applicazioni nei sistemi mobili.
La struttura dati GML, essendo XML-compatibile, si presta ad essere facilmente trasformata in qualsiasi altro formato di documento (XHTML, SVG, VML, X3D, Voice, ...) a tutto vantaggio dell'interoperabilità e dell'accessibilità.
Essendo questa tecnologia tuttora in evoluzione, questo lavoro si propone di offrire una panoramica su alcuni aspetti dello stato dell'arte, una valutazione dei vantaggi e degli svantaggi connessi e di presentare una applicazione dimostrativa sull'uso della tecnologia stessa nel settore della costruzione ed interrogazione di mappe tematiche.
L'applicazione si pone come obiettivo di elaborare documenti GML restituendoli in forma grafica ed "interrogabili" dall'utente; è stato pertanto individuato un insieme di funzionalità di base, capace di gestire sia gli aspetti puramente grafici (zoom, pan, simboli e stili, ...) che quelli più specificatamente geografici e tematici (multitematismi, classificazione ed aggregazione per attributi, ...). Tali funzionalità sono realizzate sul client Microsoft Explorer in ambiente ECMA Script, utilizzando librerie che implementano primitive di base operanti su strutture dati GML e SVG (come da raccomandazioni del W3C).

2. GIS & XML

Una serie di problemi collegati alla interoperabilità dei dati (relativi al formato, al modello spaziale, a software GIS e applicazioni GIS diverse, a sistemi GIS diversi connessi in Internet,...) sono da tempo alla base di discussioni nel mondo dell'informatica.
Un contributo alla loro soluzione è dato dall'eXtensible Mark-up Language (XML) per memorizzare e rappresentare i dati in modo estensibile e flessibile, secondo un appropriato paradigma, indipendente dal produttore dei dati. Esso rappresenta infatti uno strumento per codificare i dati in modo testuale e può combinare insieme un'ampia varietà di tipi di dati, inclusi testi, grafici, audio, ecc., compresa quindi l'informazione spaziale, amplificandone il valore e l'accessibilità. La tecnologia XML è oggi estremamente diffusa ed è inclusa nei browser dei personal computer. Mentre l'HTML racchiude insieme il contenuto e la sua presentazione, l'XML li separa nettamente; il suo standard di codifica si riferisce non alla struttura del documento ma esclusivamente a quella dei dati. Inoltre, a differenza dell'HTML, l'XML fornisce un meccanismo di collegamento (linking) a più risorse (singoli elementi XML o anche frammenti), con associazioni complesse, anche in modo bidirezionale. Poiché l'XML separa la presentazione dal contenuto, sono state sviluppate tecnologie XML di trasformazione per un'ampia varietà di devices, dal desktop computer al palmare, al wireless PDA.

L'Open GIS Consortium opera nel contesto di definire un formato neutrale (non legato ai produttori di software GIS, anche se questi sono i principali attori), ed uno dei più recenti sviluppi è stata la definizione delle specifiche del Geography Mark-up Language (GML).

2.1 Il Geography Markup Language (GML)

Il GML è un linguaggio per rappresentare i dati geografici ed il loro significato basato su XML. I dati geografici si riferiscono ad una rappresentazione del mondo in termini spaziali che è indipendente da qualsiasi particolare visualizzazione dei dati medesimi. Il GML ha la propria struttura per codificare i dati spaziali e le loro associazioni, che può essere estesa anche per modellare uno specifico contesto applicativo. In generale, il GML ha la capacità di combinare ed associare dati geospaziali con altri tipi di dati (espressi XML o GML), portando a soluzione il problema legato alla integrazione dei dati eterogenei (attualmente di formati e modelli diversi).
Al suo livello più basso, il GML fornisce un metodo per rappresentare gli elementi spaziali (feature) in modo testuale all'interno di un documento XML. Tali documenti sono semplici da decifrare e possono essere estesi per rappresentare le relazioni non solo con altri elementi spaziali, ma anche con attributi non-spaziali ed entità complesse (sfruttando le elevate capacità di "linking" dell'XML). Il GML deve essere "styled" per poter essere "presentato", generalmente come mappa o immagine, ma anche come testo o addirittura voce.
L'implementazione del GML consente inoltre di fornire un mezzo per il trasferimento e la memorizzazione dei dati in ambiente Internet. Per questo esso mette a disposizione un'ampia varietà di rappresentazioni di elementi spaziali, consente di rimuovere il meccanismo di presentazione dai dati e permette di mettere in relazione gli elementi spaziali con i rispettivi attributi, per facilitare la modellazione delle relazioni spaziali.

2.2 Gli oggetti geografici codificati secondo le specifiche GML

Il GML 2.0 è in grado di rappresentare l'informazione geografica in formato vettoriale. Una mappa può essere considerata come un insieme di strati informativi vettoriali (layer) e in GML viene rappresentata tramite l'elemento *featureCollection* che contiene le informazioni relative a tutti i layer. Al suo interno sono presenti gli elementi *boundedBy* e *featureMember*, il primo per rappresentare il rettangolo (elemento *Box*) di contenimento di tutti gli elementi dei vari layer, il secondo per definire le proprietà geometriche ed alfanumeriche dei singoli layer. Gli oggetti geografici contenuti all'interno dei layer sono rappresentati dall'elemento *Feature*. Questo è composto di due parti, una per le proprietà geometriche (*geometryProperty*) e una per gli attributi (*Property*). All'interno dell'elemento *geometricProperty* sono contenuti gli oggetti geografici rappresentati da punti (*Point*), linee (*LinearRing*) e aree (*Polygon*), ciascuno dei quali definito dall'elemento *coordinates* che specifica le coordinate dei singoli oggetti. Il sistema di coordinate geografico, rispetto al quale sono espresse le coordinate degli elementi, è indicato tramite l'attributo *srsName*. E' possibile definire questo attributo per ogni singolo elemento GML che contiene al suo interno l'indicazione di coordinate: questo consente di inserire oggetti geografici espressi in diversi sistemi di coordinate, all'interno dello stesso documento o addirittura dello stesso layer.
Per quanto riguarda gli attributi, nelle specifiche GML 2.0, non viene indicato un modo unico per rappresentarli, ma possono essere inseriti all'interno di un documento in un qualsiasi formato XML. Nell'approccio adottato è stato scelto di inserire gli attributi all'interno dell'elemento `Property`, specificandone il nome, il tipo ed il valore (vedi struttura nell'esempio seguente).

```
<?xml version="1.0" encoding="iso-8859-1"?>
<featureCollection>
   <boundedBy>
      <Box srsName="EPSG:XXXX">
         <coordinates>
         x1,y1 x2,y2
         </coordinates>
      </Box>
   </boundedBy>
<featureMember fid="PointLayer">
   <Feature fid=" PointFeature1">
      <Property name="AttrName" type="AttrType">
         AttributeValue
      </Property>
      <geometryProperty gid="PointGeometryId">
         <Point srsName="EPSG:XXXX">
            <coordinates>
            x1,y1
            </coordinates>
         </Point>
      </geometryProperty>
   </Feature>
</featureMember>
<featureMember fid="PolygonLayer">
   <Feature fid=" PolyFeature1">
      <Property name="AttrName" type="AttrType">
      AttributeValue
      </Property>
      <geometryProperty gid="PolyGeometryId">
      <Polygon srsName="EPSG:XXXX">
         <outerBoundaryIs>
            <LinearRing>
            <coordinates>
            x1,y1 .. xn,yn .. x1,y1
            </coordinates>
            </LinearRing>
         </outerBoundaryIs>
      </Polygon>
      </geometryProperty>
   </Feature>
</featureMember>
<featureMember fid="LineLayer">
   <Feature fid=" LineFeature1">
      <Property name="AttrName" type="AttrType">
         AttributeValue
      </Property>
      <geometryProperty gid="LineGeometryId">
         <LineString srsName="EPSG:XXXX">
            <coordinates>
            x1,y1 .. xn,yn
            </coordinates>
            </LineString>
      </geometryProperty>
   </Feature>
</featureMember>
</featureCollection>
```

2.3 La rappresentazione grafica di un documento GML

Il GML ha come finalità primaria quella di rappresentare il contenuto dei dati geografici. Esso può anche essere usato per rappresentare tali dati come mappe, mediante l'utilizzo di uno strumento di rendering che interpreti i dati GML. Per questo basta "ricodificare" gli elementi GML in un formato che può essere interpretato da un display grafico in un web browser (map styling). Questo comporta la traduzione dei contenuti GML mediante simboli grafici, stili di linee, riempimento di aree e volumi, e spesso anche la trasformazione della geometria del dato in quella della presentazione visiva. L'Extensible Stylesheet Language Transformation (XSLT) può essere uno strumento per eseguire l'operazione di "styling"; fa uso dell'Extensible Stylesheet Language (XSL), un linguaggio per la trasformazione di un documento XML (ad es. GML) in un altro documento XML (ad es. SVG) secondo regole di trasformazione definite specificatamente. Generalmente tale processo (graphical rendering) trasforma i dati GML in un formato grafico XML. Alcuni noti formati grafici di visualizzazione sono:

- Scalable Vector Graphics (SVG) del Consorzio W3;
- Vector Markup Language (VML);
- Web 3D del Consorzio X3D.

Attualmente esiste una varietà di graphical render per i vari formati grafici XML, sia nativi nei web browser, sia come plug-in per vari browser che come visualizzatori stand-alone o librerie di funzioni.

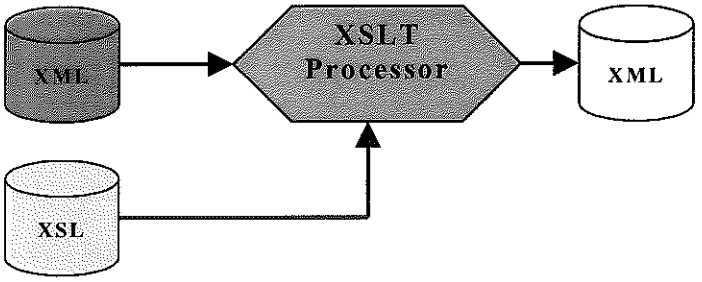
L'operazione di map styling può essere eseguita sia sul server che sul client, mediante Map Style Sheets completamente portabili da server a client e da client a client.

2.4 XSL

L'Extensible Stylesheet Language (XSL) è un linguaggio per esprimere regole di trasformazione (*stylesheet*). Consiste di tre parti:

1. *XSL Trasformation (XSLT)*: un linguaggio per la trasformazione di documenti XML;
2. *XML Path Language (Xpath)*: un linguaggio di espressione usato dall'XSLT per accedere o indirizzare parti di un documento XML (è anche usato nelle specifiche XML Linking);
3. *XSL Formatting Objects*: un vocabolario XML per specificare una semantica di formattazione.

Uno stylesheet XSL specifica la presentazione di una classe di documenti XML descrivendo come una istanza della classe è trasformata in un documento XML che usa il vocabolario di formattazione. Il processo di trasformazione converte la struttura di un documento XML in un'altra struttura di documento XML. Tramite questo linguaggio è possibile convertire un documento XML da una struttura semantica a una struttura di visualizzazione (ad esempio la trasformazione di un documento XML in documento HTML o in un documento SVG) oppure in un'altra struttura semantica (ad esempio la conversione di un documento XML in un documento XML strutturato in maniera differente).



Un esempio di trasformazione XSL (*stylesheet*) è illustrato nel paragrafo 2.6 ed in particolare riguarda la trasformazione di un documento GML in un documento SVG.

Analizziamo adesso più in dettaglio la struttura di uno stylesheet XSL. Tale documento è costituito da un insieme di modelli (racchiusi all'interno dei tag `template`) che descrivono le trasformazioni da applicare al documento di origine per ottenere l'output desiderato. Questi template vengono applicati in maniera ricorsiva alle parti del documento di origine indicate nell'attributo `match` del tag `template`.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="xml" omit-xml-declaration="yes" indent="yes"/>
<xsl:template match="/">
   <!—Regole per la gestione dell'elemento root-->
   . . . . . .

   . . . . . .
</xsl:template>
</xsl:stylesheet>
```

La prima riga indica che si tratta di un documento XML specificandone la versione e il tipo di encoding. Il tag `xsl:stylesheet` costituisce la radice del documento ed al suo interno viene definito l'insieme di modelli che vengono applicati alla struttura del documento di origine per generare il documento di output. Il tag `xsl:output` permette di definire alcuni parametri del documento che sarà il risultato della trasformazione XSL. Il tag `xsl:template` definisce il modello (regole) di trasformazione che verrà applicato alla radice (`match="/"`) del documento di input.

2.5 SVG

SVG (Scalable Vector Graphics) è un linguaggio di grafica vettoriale bidimensionale basato su XML. In SVG possono essere definiti tre tipi di oggetti grafici: forme vettoriali (punti, linee, poligoni), immagini e testo. Le caratteristiche principali del formato grafico SVG sono:

⇨ formato grafico vettoriale: consente operazioni di "scale" e "zoom"; un'immagine può essere visualizzata a qualsiasi risoluzione ed essere ingrandita o ridotta in qualsiasi sua parte senza

che si abbia alcuna perdita di qualità. Le dimensioni del file risultano ridotte rispetto ai formati grafici di tipo raster (Gif, Jpeg, etc.);

- ⇨ file "plain text": un file SVG è un file testuale è può essere facilmente letto o modificato da numerosi tool di editing alfanumerico;
- ⇨ scripting ed animazioni: SVG supporta un linguaggio di scripting (ECMA Script) attraverso il quale è possibile rendere interattivi gli elementi SVG o creare degli effetti di animazione;
- ⇨ standard aperto: SVG è uno standard aperto sviluppato dal Consorzio W3;
- ⇨ applicazione XML: SVG è un'applicazione XML e come tale è in grado di offrire tutti i vantaggi tipici dell'XML, quali ad esempio:
  - interoperabilità;
  - internazionalizzazione (supporta la codifica Unicode);
  - supporto da parte di numerosi tool di sviluppo;
  - facilità di manipolazione attraverso API standard come ad esempio le Document Object Model (DOM) API (definite dal W3C per definire un metodo univoco indipendente dai linguaggi di programmazione e dalle piattaforme per l'accesso a documenti XML);
  - trasformabilità dei documenti XML attraverso XSL.

Analizziamo un esempio di file SVG in modo da mostrare più in dettaglio la struttura di un documento SVG. In questo esempio è rappresentato, all'interno di una cornice blu, un rettangolo giallo con bordo blu (Fig. 2.5.1.a).

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20010904//EN"
   "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
<svg width="12cm" height="4cm" viewBox="0 0 1200 400" xmlns="http://www.w3.org/2000/svg">
   <rect x="1100" y="100" width="50" height="200" fill="yellow" stroke="navy" stroke-width="5" />
   <circle cx="800" cy="200" r="100" fill="red" stroke="blue" stroke-width="5" />
   <ellipse cx="800" cy="200" rx="250" ry="100" fill="yellow" stroke="black" stroke-width="5"  />
   <line x1="100" y1="200" x2="200" y2="200" stroke="black"  stroke-width="5" />
   <polyline fill="none" stroke="blue" stroke-width="5"
      points="50,375 150,375 150,325 250,325 250,375 350,375 350,325 450,325 450,375
         550,375 550,325 650,325 650,375 750,375 750,325 850,325 850,375
         950,375 950,325 1050,325 1050,375 1150,375"/>
   <polygon fill="red" stroke="blue" stroke-width="5"
      points="350,75  379,161 469,161 397,215 423,301 350,250 277,301 303,215
         231,161 321,161" />
   <path d="M500,100 C575,10 875,10 800,50Z" fill="blue" stroke="navy" stroke-width="5" />
</svg>
```

Fig. 2.5.1.a – Documento SVG rappresentante varie shape grafiche

La prima riga del documento SVG indica che si tratta di un documento XML e specifica alcuni parametri quali ad esempio la versione e l'encoding. La seconda riga indica il tipo di documento e il file contenente la grammatica per il file SVG.

Il tag `svg` è la radice del documento XML ed i suoi attributi specificano alcuni parametri che riguardano l'intero documento SVG, quali ad esempio la dimensione e i namespace utilizzati all'interno del documento XML.

Il tag `rect` rappresenta un rettangolo ed i suoi attributi specificano alcuni parametri quali la posizione (`x`, `y`), la grandezza (`width` e `height`) e la colorazione del rettangolo (`fill`, `stroke`, `stroke-width`).

Il tag `circle` rappresenta un cerchio di raggio `r` , centrato in `cx` e `cy` e lo stile di rappresentazione espresso dagli attributi `fill`, `stroke`, `stroke-width`.
Il tag `ellipse` rappresenta un'ellisse ed i suoi attributi specificano il centro (`cx`, `cy`), le dimensioni dei raggi (`rx`, `ry`) e lo stile di rappresentazione (`fill`, `stroke`, `stroke-width`).
Il tag `line` rappresenta una linea e i suoi attributi `x1,y1` e `x2,y2` indicano le coordinate degli estremi di questa linea. Lo stile di rappresentazione è espresso dagli attributi `stroke`, `stroke-width`.
Il tag `polyline` rappresenta una spezzata e i suoi attributi specificano le coordinate dei punti che costituiscono la spezzata (`points`) e lo stile di rappresentazione (`fill`, `stroke`, `stroke-width`).
Il tag `polygon` rappresenta un poligono costituito dai punti indicati all'interno dell'attributo points, mentre gli altri attributi esprimono lo stile di visualizzazione.
Il tag `path` esprime il bordo di uno shape e l'attributo d contiene le istruzioni utili a rappresentare la figura geometrica. Lo stile è indicato dagli attributi `stroke`, `fill`, `stroke-width`.

Shape Tag	Descrizione
`<rect>`	Rettangolo
`<circle>`	Cerchio
`<ellipse>`	Ellisse
`<line>`	Linea
`<polyline>`	Spezzata
`<polygon>`	Poligono
`<path>`	Bordo di una Shape

Tab. 2.5.1.a – Principali Shape tag SVG

2.6 Trasformazione da GML a SVG

In questo paragrafo è mostrato come le entità geografiche punto, linea, area sono codificate GML e come vengono successivamente trasformate nelle equivalenti entità codificate SVG (Tab. 2.6.a).
Il file SVG (visualizzato dal web browser) è ottenuto applicando una trasformazione XSL al documento GML da parte del XSLT Processor.

Entità Geogr.	Tag GML	Tag SVG
Punto	Point	circle, rect, etc.
Linea	LineString	polyline
Area	LinearRing	path

Tab. 2.6.a – Corrispondenza tra shape GML e SVG per le entità geografiche

2.6.1 Trasformazione di un punto

In figura 2.6.1.a è riportata la parte di documento GML che descrive l'elemento puntuale `P1`. Le proprietà geografiche di questo elemento sono specificate dal tag `geometryProperty`, che contiene l'elemento GML che rappresenta il punto (`Point`). Le coordinate relative al punto sono espresse dal tag `coordinates`.

```
....
....
<Feature fid="P1" featureType="Punto">
  <geometryProperty gid="p1-point">
    <Point srsName="EPSG:XXXX">
      <coordinates>
        9650.267,5334240.557
      </coordinates>
    </Point>
  </geometryProperty>
</Feature>
....
....
```

Fig. 2.6.1.a – Rappresentaz. GML di un punto

In figura 2.6.1.b è riportata la parte di codice XSL che è applicata per la trasformazione. Il template rappresentato dal tag `xsl:template` descrive il modello di trasformazione da applicare all'elemento `Point` del documento GML. Il tag `xsl:element` crea l'elemento SVG `circle` che rappresenta il punto (mediante un cerchio) ed i suoi attributi (ovvero l'id, lo stile di rappresentazione, le coordinate del centro ed il raggio).

```
....
....
<xsl:template match="Point">
  <xsl:variable name="clist" select="./coordinates"/>
  <xsl:variable name="cpair" select="normalize-space($clist)" />
  <xsl:variable name="x">
    <xsl:value-of select="math:tcoordx($cpair)"/>
  </xsl:variable>
  <xsl:variable name="y">
    <xsl:value-of select="math:tcoordy($cpair)"/>
  </xsl:variable>
    <xsl:element name="circle" namespace="">
      <xsl:attribute name="id">
        <xsl:value-of select="normalize-space(ancestor-or-self::Feature/./@fid)"/>
      </xsl:attribute>
      <xsl:attribute name="cx"><xsl:value-of select="$x"/></xsl:attribute>
      <xsl:attribute name="cy"><xsl:value-of select="$y"/></xsl:attribute>
      <xsl:attribute name="r">
        <xsl:value-of select="math:width('c')"/>
      </xsl:attribute>
      <xsl:attribute name="style">
        fill:red;stroke:black;stroke-width:0.25;stroke-dasharray:0
      </xsl:attribute>
    </xsl:element>
</xsl:template>
.....
.....
```

Fig. 2.6.1.b – Codice XSL per la trasformazione di un punto

I valori delle coordinate del centro del cerchio, contenute all'interno delle variabili XSL `x` e `y`, sono ottenuti come risultato di due funzioni (`tcoordx` e `tcoordy`). Queste funzioni effettuano le necessarie trasformazioni tra il sistema di coordinate nel quale sono espresse le coordinate nel documento GML e quello utile per la visualizzazione della mappa. Anche il valore del raggio del cerchio è il risultato di una funzione (`width`) ed è ottenuto in modo tale che il suo valore sia proporzionale alla dimensione generale della mappa.

In figura 2.6.1.c è riportata la parte di documento SVG che descrive l'elemento puntuale `P1`, derivante dalla trasformazione XSLT applicata al documento GML. Come si vede, è stato creato un elemento `circle` con i seguenti attributi: `Id` che rappresenta l'identificatore dell'elemento geografico; `style` che indica lo stile scelto per la rappresentazione e `d` che racchiude le istruzioni per disegnare il poligono espresse secondo la sintassi SVG.

```
....
....
<circle
   id="P1"
   cx="354.75347249999976"
   cy="139.01045249995775"
   r="0.5"
   style="fill:red;stroke:black;stroke-width:0.25;stroke-dasharray:0"
/>
....
....
```

Fig. 2.6.1.c – Rappresentazione SVG di un punto

2.6.2 Trasformazione di una linea

In figura 2.6.2.a è riportata la parte di documento GML che descrive l'elemento lineare `A1`. Le proprietà geografiche di questo elemento sono specificate dal tag `geometryProperty`, che contiene l'elemento GML che rappresenta la linea (`LineString`). Le coordinate relative ai punti che costituiscono la linea sono espresse dal tag `coordinates`.

```
....
....
<Feature fid="L1" featureType="Linea">
  <geometryProperty gid="L1-line">
    <LineString srsName="EPSG:XXXX">
      <coordinates>
        9685.267,5334248.557
        9565.367,5334270.186
        9506.480,5334159.730
      </coordinates>
    </LineString>
  </geometryProperty>
</Feature>
.....
.....
```

Fig. 2.6.2.a – Rappresentazione GML di una linea

In figura 2.6.2.b è riportata la parte di codice XSL che è applicata per la trasformazione. Il template rappresentato dal tag `xsl:template` descrive il modello di trasformazione da applicare all'elemento `LineString` del documento GML. I valori descritti dal tag `coordinates` vengono memorizzati in una variabile e successivamente elaborati dalla funzione `tlista`.
Questa funzione effettua le necessarie trasformazione tra il sistema di coordinate nel quale sono

```
....
....
<xsl:template match="LineString">
   <xsl:variable name="width" select="math:width('l')"/>
   <xsl:variable name="parametri">
      stroke:blue;fill:none;stroke-dasharray:0;stroke-width:
   </xsl:variable>
   <xsl:variable name="clist" select="./coordinates"/>
   <xsl:variable name="tclist" select="normalize-space($clist)" />
      <xsl:element name="polyline">
      <xsl:attribute name="id">
         <xsl:value-of select="normalize-space(ancestor-or-self::Feature/./@fid)"/>
      </xsl:attribute>
      <xsl:attribute name="style">
         <xsl:value-of select="concat($parametri,$width)"/>
      </xsl:attribute>
      <xsl:variable name="listat" select="math:tlista($tclist)" />
      <xsl:attribute name="points">
         <xsl:value-of select="$listat"/>
      </xsl:attribute>
      </xsl:element>
</xsl:template>
....
....
```

Fig. 2.6.2.b – Codice XSL per la trasformazione di una linea

espresse le coordinate nel documento GML e quello utile per la visualizzazione della mappa. Il tag `xsl:element` crea l'elemento SVG `polyline` che rappresenta la linea ed i suoi attributi (ovvero l'id, lo stile di rappresentazione e le coordinate). Il valore dell'attributo relativo allo spessore della linea è determinato in maniera dinamica dalla funzione width che restituisce un valore proporzionale alla dimensione generale della mappa.
In figura 2.6.2.c è riportata la parte di documento SVG che descrive l'elemento lineare `L1`,

```
....
....
<polyline
   id="L1"
   style="stroke:blue;fill:none;stroke-dasharray:0;stroke-width:0.5"
   points="404.3659724999997,127.67045249995777
      234.39354749999995,97.01134500034851,150.9353999999994,253.58272499936632"
/>
....
....
```

Fig. 2.6.2.c – Rappresentazione SVG di una linea

derivante dalla trasformazione XSLT applicata al documento GML.
Come si vede è stato creato l'elemento `polyline` con i seguenti attributi: `Id` che rappresenta l'identificatore dell'elemento geografico; `style` che indica lo stile scelto per la rappresentazione e `points` che contiene i valori delle coordinate dei punti che formano la linea.

2.6.3 Trasformazione di un poligono

In figura 2.6.3.a è riportata la parte di documento GML che descrive l'elemento areale `A1`. Le proprietà geografiche di questo elemento sono specificate dal tag `geometryProperty` ed i suoi confini esterni sono rappresentati dal tag `LinearRing`. Il tag `coordinates` contiene le coordinate relative ai punti della linea chiusa che racchiude l'area rappresentata da `A1`.

```
.....
.....
<Feature fid="A1" featureType="Area">
  <geometryProperty gid="p1Boundary">
    <Polygon srsName="EPGG:XXXX">
      <outerBoundaryIs>
        <LinearRing>
          <coordinates>
            9557.743,5334329.953 9566.831,5334326.288
            9635.264,5334302.503 9631.897,5334297.010
            9588.745,5334226.630 9535.357,5334245.166
            9526.168,5334248.574 9557.743,5334329.953
          </coordinates>
        </LinearRing>
      </outerBoundaryIs>
    </Polygon>
  </geometryProperty>
</Feature>
.....
.....
```

Fig. 2.6.3.a – Rappresentaz. GML di un poligono

In figura 2.6.3.b è riportata la parte di codice XSL che è applicata per la trasformazione. Il template rappresentato dal tag

```
.....
.....
<xsl:template match="Polygon">
  <xsl:variable name="clist" select="./coordinates"/>
  <xsl:variable name="coord" select="math:tlista($clist)"/>
  <xsl:variable name="start" select="concat('M',$coord)"/>
  <xsl:variable name="start2" select="translate($start,' ','L')"/>
  <xsl:variable name="all" select="concat($start2,'Z')"/>
  <xsl:element name="path" >
    <xsl:attribute name="id">
      <xsl:value-of select="normalize-space(ancestor-or-self::Feature/./@fid)"/>
    </xsl:attribute>
    <xsl:attribute name="style">
      stroke:red;fill:lightyellow;stroke-dasharray:0;
      stroke-width:0.5;display:inline;opacity:1
    </xsl:attribute>
    <xsl:attribute name="d">
      <xsl:value-of select="$all"/>
    </xsl:attribute>
  </xsl:element>
</xsl:template>
.....
.....
```

Fig. 2.6.3.b – Codice XSL per la trasformazione di un poligono

`xsl:template` descrive il modello di trasformazione da applicare all'elemento `Polygon` del documento GML. I valori descritti dal tag `coordinates` vengono memorizzati in una variabile e successivamente elaborati dalla funzione `tlista`. Questa funzione effettua le necessarie trasformazione tra il sistema di coordinate nel quale sono espresse le coordinate nel documento GML e quello utile per la visualizzazione della mappa. Queste coordinate sono ulteriormente elaborate per essere espresse secondo le regole della grammatica SVG. Il tag `xsl:element` crea l'elemento SVG `path` che rappresenta il poligono ed i suoi attributi (ovvero l'id, lo stile di rappresentazione e le coordinate).

In figura 2.6.3.c è riportata la parte di documento SVG che descrive l'elemento areale `A1`, derivante dalla trasformazione XSLT applicata al documento GML.

```
....
....
<path
   id="A1"
   style="stroke:red;fill:lightyellow;stroke-dasharray:0; stroke-width:0.5;display:inline;
         opacity:1"
   d="M223.60070250000058,12.291622500359096L236.48294250000018,17.48676000041189L333.48671999
         999885,51.20199750062312L328.7139975000012,58.98832500031682L267.54603750000114,158.7
         5197500015844L191.86854749999995,132.4488450003485L178.84313999999952,127.64635499996
         831,223.60070250000058,12.291622500359096Z"
/>
....
....
```

Fig. 2.6.3.c – Rappresentazione SVG di un poligono

Come si vede, è stato creato un elemento `path` con i seguenti attributi: `Id` che rappresenta l'identificatore dell'elemento geografico; `style` che indica lo stile scelto per la rappresentazione e `d` che racchiude le istruzioni per disegnare il poligono espresse secondo la sintassi SVG.

3. GML Data Explorer

L'applicazione GML Data Explorer permette la visualizzazione e l'analisi dei dati spaziali codificati mediante il GML. Il programma relativo (`gmle.html`) fa uso di routines ECMA Script ed utilizza le funzionalità di visualizzazione del plugin SVG della Adobe, realizzate interamente lato client mediante il browser web Microsoft Internet Explorer.
Sono implementate le seguenti funzionalità:

- visualizzazione della mappa;
- gestione della visualizzazione della mappa: pan, zoom, simboli, colori, stili, etichette, etc.;
- visualizzazione degli strati informativi (layer) mediante organizzazione gerarchica;
- selezione del layer attivo (oggetto delle operazioni successive);
- classificazione dei dati in base agli attributi e successiva rappresentazione grafica;
- descrizione delle proprietà dei layer (attributi);
- descrizione degli elementi selezionati;
- visualizzazione dei dati sorgente GML.

Le funzionalità di *GML Data Explorer* sono messe a disposizione dell'utente come tools accessibili dall'interfaccia grafica mediante bottoni.

3.1 Architettura del GML Data Explorer

L'architettura dell'applicazione, mostrata in Fig. 3.1, evidenzia come tutte le funzionalità di GML Data Explorer siano realizzate interamente lato client, sfruttando le capacità del *Plugin SVG* (Adobe), del linguaggio *JavaScript* e del *Parser XML*.
I documenti GML e XSL possono indifferentemente risiedere sia sul disco locale del client che sul disco di un web server remoto.

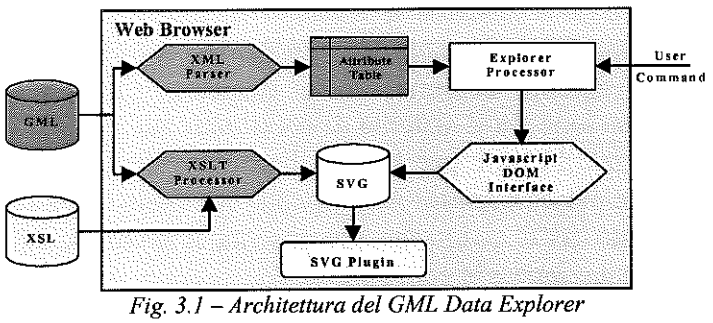


Fig. 3.1 – Architettura del GML Data Explorer

I moduli principali del sistema sono:

- **XSLT processor**: trasforma i dati geografici da documento GML a documento SVG. Tale trasformazione viene eseguita applicando le regole contenute nel documento XSL;

- **XML parser** : analizza il documento GML ed estrae i dati riguardanti gli attributi delle feature geografiche. I dati estratti vengono inseriti all'interno della *Attribute Table*, una struttura dati organizzata a tabella, che verrà utilizzata per effettuare le operazioni di classificazione. Il parser utilizzato è il *Microsoft XMLDOM parser*;
- **Explorer processor** : gestisce le funzioni di classificazione dei dati. Analizza gli attributi degli oggetti geografici, contenuti nella *Attribute Table,* e, in base al tipo di classificazione ed ai parametri scelti dall'utente, rappresenta il risultato dell'operazione di classificazione modificando il documento SVG. Tali modifiche vengono effettuate sfruttando la *JavaScript DOM Interface.*

3.2 Funzionamento

Il programma (attivabile alla URL http://webgisserver/gmlexplorer/gmle.html) istanzia due *Microsoft ActiveXObject*[1] (embedded nel browser *Microsoft Explorer)* di tipo XMLDOM dove verranno memorizzati il DOM del documento GML e quello del documento XSL. Successivamente viene attivato il processore XSLT, contenuto all'interno del browser Internet Explorer, per applicare la trasformazione XSL al documento GML. Il risultato di questa trasformazione è la creazione di un altro oggetto ActiveX di tipo XMLDOM contenente il DOM del documento SVG che rappresenta graficamente i dati GML. Questo DOM viene fatto processare dal *SVG Plugin*, precedentemente integrato nel browser, in modo da visualizzare il risultato della trasformazione XSL. Tale risultato è costituito da un documento SVG contenente la rappresentazione di tutti gli oggetti geografici contenuti nel vari layer.
Effettuata la trasformazione XSL, il documento GML viene ulteriormente analizzato da un *XML Parser* (Microsoft XMLDOM parser) al fine di estrarre gli attributi degli oggetti geografici contenuti nel layer SVG attivo. Questi dati sono inseriti all'interno della *Attribute Table*, ovvero una struttura dati organizzata a matrice per effettuare le operazione di classificazione. Questa tabella costituisce una sorta di buffer dati utilizzato per effettuare le analisi necessarie alle operazioni di classificazione. L'utilizzo di questo buffer permette di velocizzare le operazioni di lettura dei dati limitando sia l'accesso al disco in lettura sia un ulteriore parsing del documento GML alla ricerca di dati già presenti in memoria.
Attraverso l'interfaccia di *GML Data Explorer* l'utente può scegliere il tipo di classificazione, l'attributo in funzione del quale eseguire la classificazione e la rappresentazione grafica interessata (colore, stile, simbolo, etc.).
Le operazioni di classificazione effettuate da GML Data Explorer sono:

- *Unique Color*: viene associato un unico colore, scelto all'interno di un gradiente di colori, a ciascun valore dell'attributo scelto per effettuare la classificazione;
- *Gradient Color*: il range dei valori possibili assunto dall'attributo di classificazione viene diviso in intervalli a cui viene associato un particolare colore. Questi intervalli sono determinati in tre modi diversi: equal interval, equal area o standard deviation;
- *Gradient Symbol*: viene associato a ciascun elemento un simbolo (rettangolo, cerchio, triangolo, etc.) con dimensione espressa dal risultato della classificazione.
- *Chart*: viene associato a ciascun elemento un grafico di tipo istogramma rappresentante i valori di alcuni attributi, scelti dall'utente, del layer attivo.

[1] ActiveX è un insieme di regole e di tecnologie sviluppato da Microsoft che permettono ad un software o ad una sua parte di interagire con altre parti esterne, indipendentemente dal linguaggio con il quale esse sono state realizzate. Attraverso l'uso dei protocolli standard, oggetti creati in un'applicazione possono essere caricati e modificati da un'applicazione differente. ActiveX deriva da altre due tecnologie Microsoft, precisamente OLE (Object Linking and Embedding) e COM (Component Object Model).

Impostati i parametri ed attivata la classificazione, il programma analizza i dati contenuti nella *Attribute Table* e determina per ciascun elemento del layer attivo il risultato dell'operazione di classificazione. Tale risultato viene quindi espresso cambiando la colorazione dell'elemento o applicando un simbolo in funzione del risultato.
Gli elementi del documento SVG e gli elementi stessi del documento GML, che contengono le informazioni relative agli attributi, sono legati attraverso l'uso di un identificatore unico, costituito dall'attributo `fid` dell'elemento GML `Feature`. Tale identificatore viene letto durante la trasformazione XSL ed assegnato all'attributo `id` dell'elemento SVG che rappresenta la feature GML. Quindi i risultati delle operazioni di classificazione vengono visualizzati modificando opportunamente gli elementi del documento SVG, ricercando l'elemento con l'identificatore appropriato.
E' possibile inoltre visualizzare le informazioni relative agli attributi degli elementi geografici sia selezionando direttamente sull'interfaccia grafica l'elemento desiderato, sia scegliendo la funzione di visualizzazione della tabella attributi, relativa al layer attivo, che consente di rappresentare sotto forma di tabella HTML la *Attribute Table*.
Tutte queste operazioni sono effettuate agendo sul documento SVG, creato dalla trasformazione XSL, e modificando le proprietà dei suoi attraverso l'interfaccia DOM del linguaggio Javascript.

4. Interfaccia utente

In Fig. 4.a è mostrata l'interfaccia principale del sistema: essa è suddivisa logicamente in finestre (*GML EXPLORER, Graphic Window, Layer, Layer Properties, Layer Operations*) contenenti bottoni e campi di selezione mediante i quali sono attivabili tutte le funzionalità.

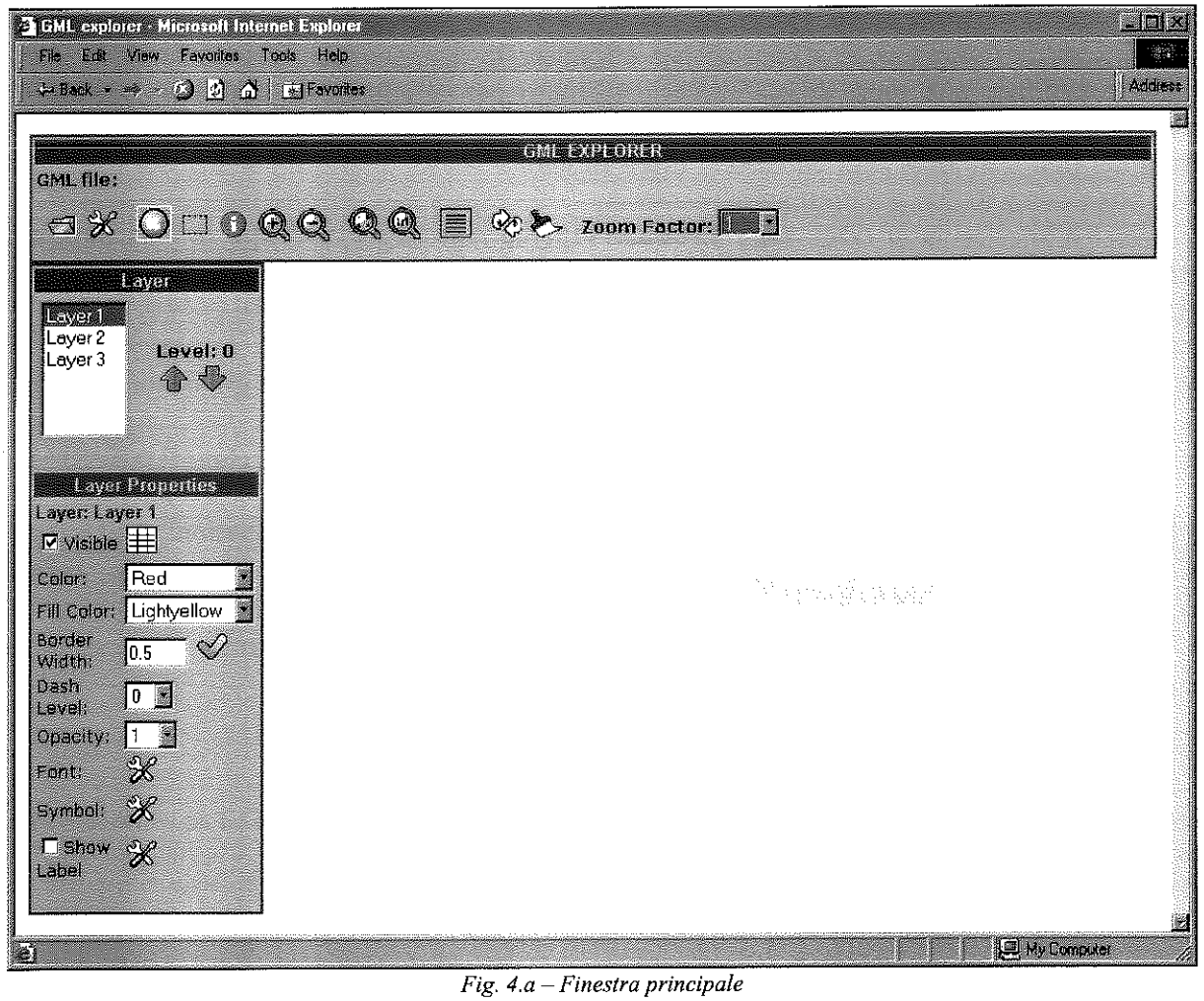


Fig. 4.a – Finestra principale

La finestra *GML EXPLORER* (Fig. 4.b) contiene un insieme di bottoni preposti al caricamento dei files GML e XSL, alle operazioni di Zoom e Pan della finestra grafica, alla visualizzazione del codice sorgente GML, ecc... .

La finestra *Layer* (Fig. 4.c) contiene l'elenco degli strati informativi presenti nel file GML precedentemente caricato: essi sono organizzati visivamente nella finestra grafica secondo una gerarchia top-bottom, manipolabile spostando in alto o in basso il layer selezionato. Le proprietà grafiche dei vari elementi presenti in ciascun layer sono gestite dalla finestra *Layer Properties* (Fig. 4.d): essa consente di selezionare un layer (layer attivo), renderlo visibile o no nella *Graphic Window*, impostare le proprietà grafiche dei vari elementi (proprie delle tipologie punto, linea, area) e visualizzare l'dentificatore degli elementi. Gli attributi del layer attivo sono visualizzabili mediante la finestra Properties Table (Fig. 4.e): essa contiene i nomi degli attributi del layer e l'elenco di tutti gli elementi che lo compongono con i rispettivi valori. Selezionando elementi (righe) nella tabella, saranno evidenziati nella *Graphic Window* i corrispondenti elementi grafici. Questa finestra è predisposta inoltre per effettuare operazioni di ordinamento degli elementi basate sugli attributi (Fig. 4.f). E' possibile infatti selezionare l'attributo interessato, il criterio di ordinamento (increasing, decreasing) ed avviare l'operazione.

	Apertura della finestra di selezione del file GML
	Apertura della finestra di selezione del file XSL
	Selezione elemento cliccato del layer attivo
	Visualizzazione della finestra contenente gli attributi dell'elemento selezionato
	Operazione di Zoom-In centrata nel punto di selezione
	Operazione di Zoom-Out centrata nel punto di selezione
	Operazione di Zoom-Fit rispetto agli elementi del layer attivo
	Operazione di Zoom-Fit rispetto all'intero documento
	Apertura della finestra di visualizzazione del listato GML
	Deselezione degli elementi selezionati
	Cancella la precedente operazione di classificazione
Zoom Factor: 0.5	Impostazione del fattore di zoom

Fig. 4.b – Toolbar principale

Layer 1 Layer 2 Layer 3	Gerarchia dello stack degli strati informativi
	Sposta in alto il layer selezionato
	Sposta in basso il layer selezionato

Fig. 4.c – Gestione layer

Layer: Layer 1	Indicazione del layer attivo
Visible	Visualizza/Nascondi il layer attivo
	Apertura finestra di visualizzazione degli attributi
Fill Color: Red	Impostazione del colore di riempimento delle aree e dei simboli
Color: Black	Impostazione del colore delle linee o dei bordi
Border Width: 0.25	Impostazione dello spessore della linea del bordo
Dash Level: 0	Impostazione del livello di tratteggiatura delle linee o dei bordi
Opacity: 1	Impostazione del livello di opacità degli elementi del layer
Font:	Apertura finestra delle proprietà dei font
Symbol:	Apertura finestra delle proprietà dei simboli
Show Label	Visualizza/Nascondi etichette ed apertura della finestra delle proprietà delle etichette

Fig. 4.d – Proprietà dei layer

La finestra *Layer Operations* (Fig. 4.g) consente di effettuare operazioni di classificazione degli elementi basate su uno o più attributi del layer attivo. L'operazione comporta l'impostazione di alcuni parametri che la definiscono; mediante la finestra di selezione delle proprietà della classificazione (Fig. 4.h), si può scegliere: il tipo di classificazione (*Unique Color, Gradient Color, Gradient Symbol, Chart*), il numero delle classi, il tipo di generazione delle classi (*Equal*

Interval,), il tipo di arrotondamento dei valori, il simbolo di rappresentazione e il gradiente di colore.

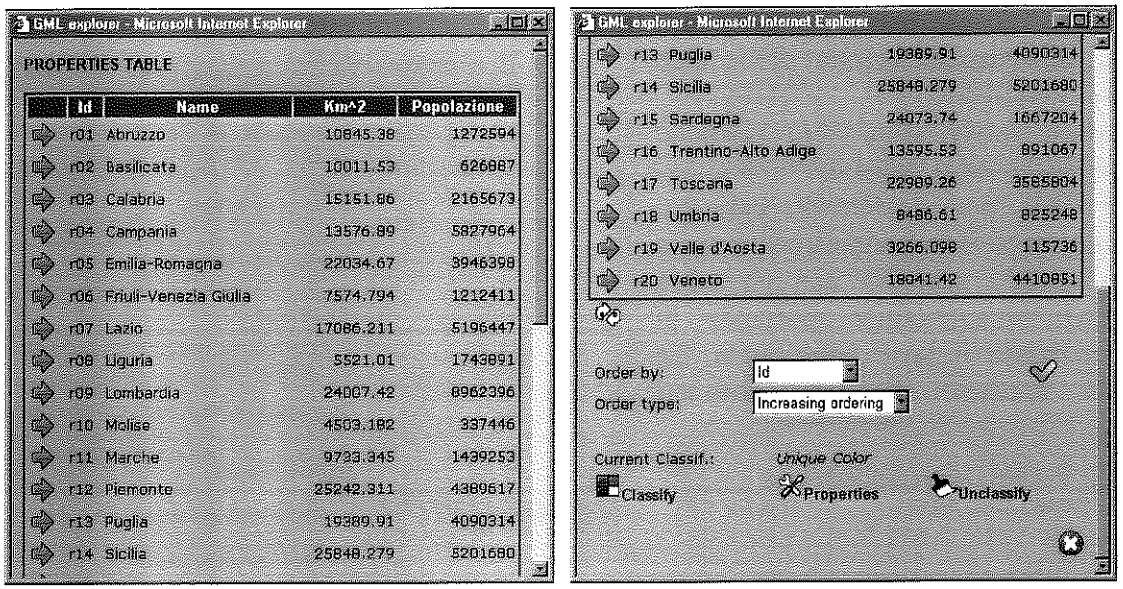


Fig. 4.e – Finestra degli attributi

	Selezione elemento rappresentato nelle riga corrispondente
	Deselezione degli elementi selezionati
	Chiusura della finestra

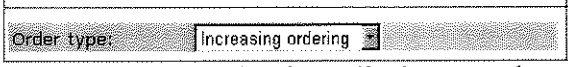
	Scelta dell'attributo su cui operare e avvio dell'ordinamento degli elementi
	Impostazione del criterio di ordinamento degli elementi

Fig. 4.f – Selezione/deselezione e ordinamento degli elementi

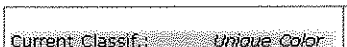
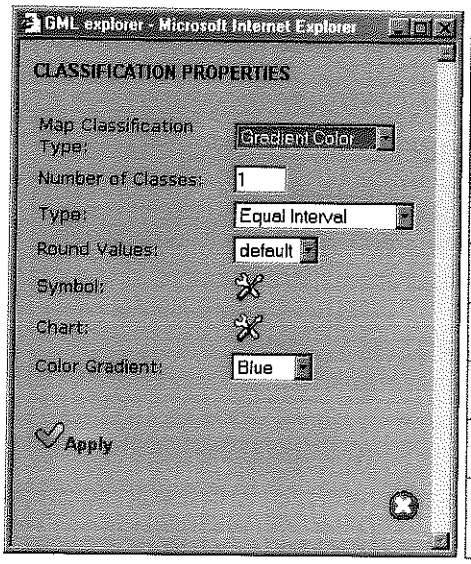
	Indicazione del tipo di classificazione attiva
	Attiva l'operazione di classificazione
	Impostazione dei parametri dell'operazione di classificazione
	Cancella la precedente operazione di classificazione

Fig. 4.g – Operazioni di classificazione



Map Classification Type: Gradient Color	Impostazione del tipo di classificazione
Number of Classes: 1	Impostazione del numero di classi
Type: Equal Interval	Impostazione del tipo di suddivisione da applicare ai valori dell'attributo
Round Values: default	Impostazione del grado di precisione
Symbol:	Apertura della finestra delle proprietà dei simboli
Chart:	Apertura della finestra di configurazione del chart
Color Gradient: Blue	Impostazione del gradiente di colore
	Applicazione delle modifiche
	Chiusura della finestra

Fig. 4.h – Proprietà della classificazione

5. Esempio di uso di GML Data Explorer

In questa sezione passiamo ad analizzare in dettaglio il funzionamento di GML Data Explorer e per far questo verrà mostrato un esempio di classificazione su dati codificati in GML. Il file utilizzato per questo esempio è *italia.xml* ed è un file contenente alcuni dati geografici (ed attributi) relativi alle regioni, ai principali fiumi e ai capoluoghi di regione dell'Italia.

Attraverso l'interfaccia mostrata in Fig. 5.a è possibile indicare il file da analizzare locale (browse delle risorse locali) o remoto (indicando un URL).

Il file viene caricato da GML Data Explorer e sottoposto alla trasformazione XSL che produrrà il documento SVG rappresentante l'Italia.

In figura 5.b viene mostrato il risultato del caricamento del file GML.

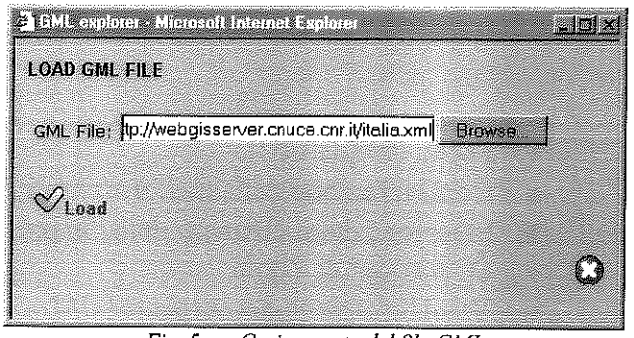


Fig. 5.a – Caricamento del file GML

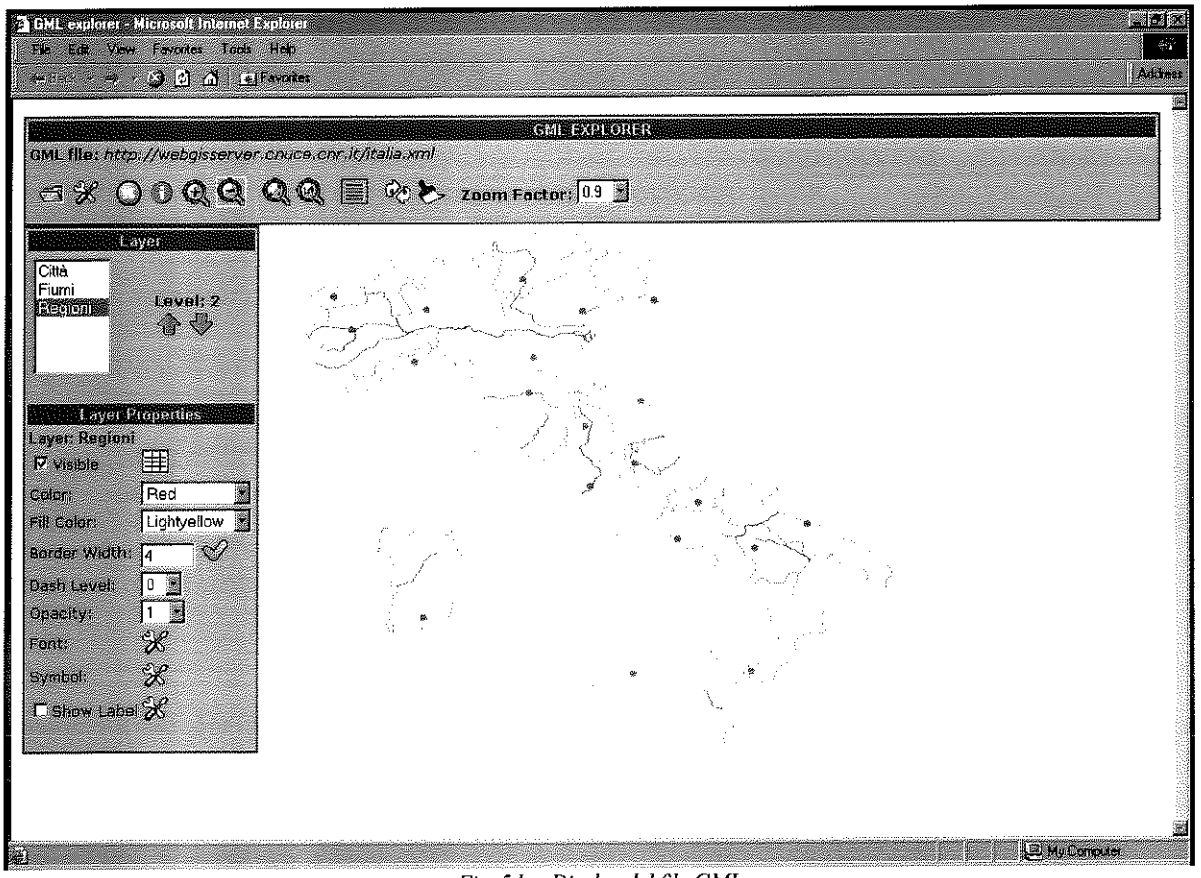


Fig. 5.b – Display del file GML

L'interfaccia grafica di GML Data Explorer è costituita da una toolbar contenente alcuni pulsanti che controllano alcune operazioni tra cui:

- Scelta del file GML
- Configurazione della trasformazione XSL
- Selezione degli elementi della mappa
- Operazioni di Zoom
- Visualizzazione del sorgente GML

Attraverso il pannello laterale è possibile gestire i layer e le proprietà di visualizzazione degli elementi della mappa. La sezione *layer* del pannello permette di aumentare o diminuire il livello del layer attivo modificandone la disposizione gerarchica rispetto agli altri strati informativi.
Nella sezione *layer properties* si trovano i comandi per modificare lo stile della rappresentazione degli elementi del layer attivo. E' possibile impostare il colore, la dimensione del bordo, il livello di opacità, il tipo di simbolo associato (solo nel caso di elemento puntuale), la visibilità e configurare le proprietà delle etichette associate agli elementi.
Inoltre attraverso l'icona presente nella sezione *layer properties*, rappresentante una tabella, è possibile aprire la finestra relativa alla parte del programma che si occupa dell'analisi degli attributi e delle operazioni di classificazione (Fig. 5.c).

Al momento dell'apertura della finestra viene attivata la funzione di analisi degli attributi del layer. Questa funzione ricerca attraverso l'utilizzo del parser XML i dati contenuti all'interno del file GML e li inserisce all'interno della Attribute Table. Ovviamente questo viene fatto

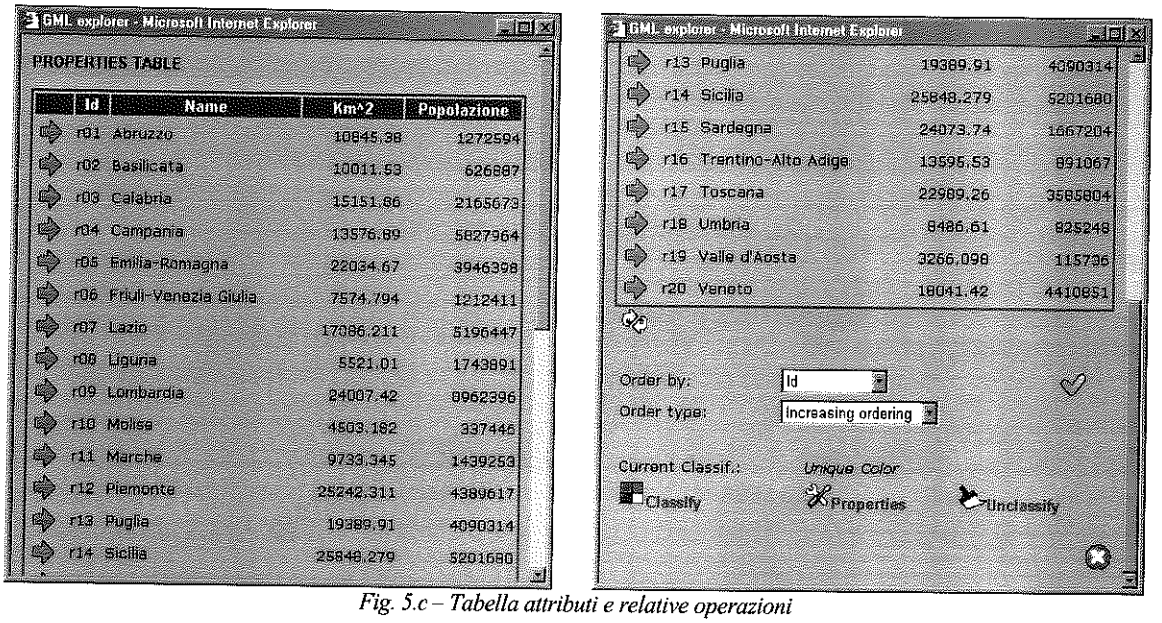


Fig. 5.c – Tabella attributi e relative operazioni

solamente se i dati relativi al layer attivo non sono già contenuti all'interno tabella buffer.
La finestra *Properties Table* contiene la tabella degli attributi relativa al layer attivo, ossia la visualizzazione della *Attribute Table*, e due gruppi di comandi utilizzati per le operazioni di classificazione sui dati. Attraverso il primo gruppo è possibile specificare l'attributo utilizzato per la classificazione e il tipo di ordinamento voluto sui valori di questo attributo. Il secondo gruppo di comandi è relativo all'operazione di classificazione vera e propria .
Tramite il pulsante *Properties* è possibile richiamare la finestra per la configurazione dei parametri della classificazione (Fig. 5.d).
All'interno di questa finestra di configurazione è possibile indicare il tipo di classificazione voluta (unique color, gradient color, gradient symbol o chart) ed i relativi parametri (range degli intervalli, precisione, tipo di definizione degli intervalli, attributi dei simboli e del grafico).
Per questo esempio, è stata scelto un tipo di classificazione *Unique Color* relativa all'attributo *Popolazione* del layer *Regioni*. La classificazione viene eseguita generando un gradiente di colori (del tipo indicato dall'utente) pari al numero di valori dell'attributo *Popolazione*. Dopo aver assegnato ad ogni valore dell'attributo un colore appartenente al gradiente, ha inizio la fase di applicazione del risultato di classificazione. In questo caso l'operazione di classificazione consiste nell'assegnare il colore relativo ad un valore dell'attributo Popolazione all'elemento SVG che rappresenta l'elemento GML a cui corrisponde quel valore di popolazione. Per ogni feature GML contenuta nel layer attivo, viene analizzato il valore dell'identificatore e il valore dell'attributo popolazione in modo da determinare il colore da associare all'elemento che rappresenta il risultato della classificazione. Una volta determinata questa associazione e dopo aver memorizzato l'informazione relativa alla colorazione

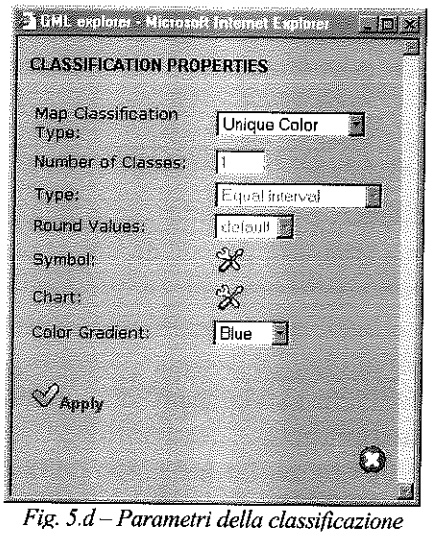


Fig. 5.d – Parametri della classificazione

dell'elemento, si procede alla modifica della colorazione dell'elemento SVG identificato dal medesimo identificatore della feature GML.

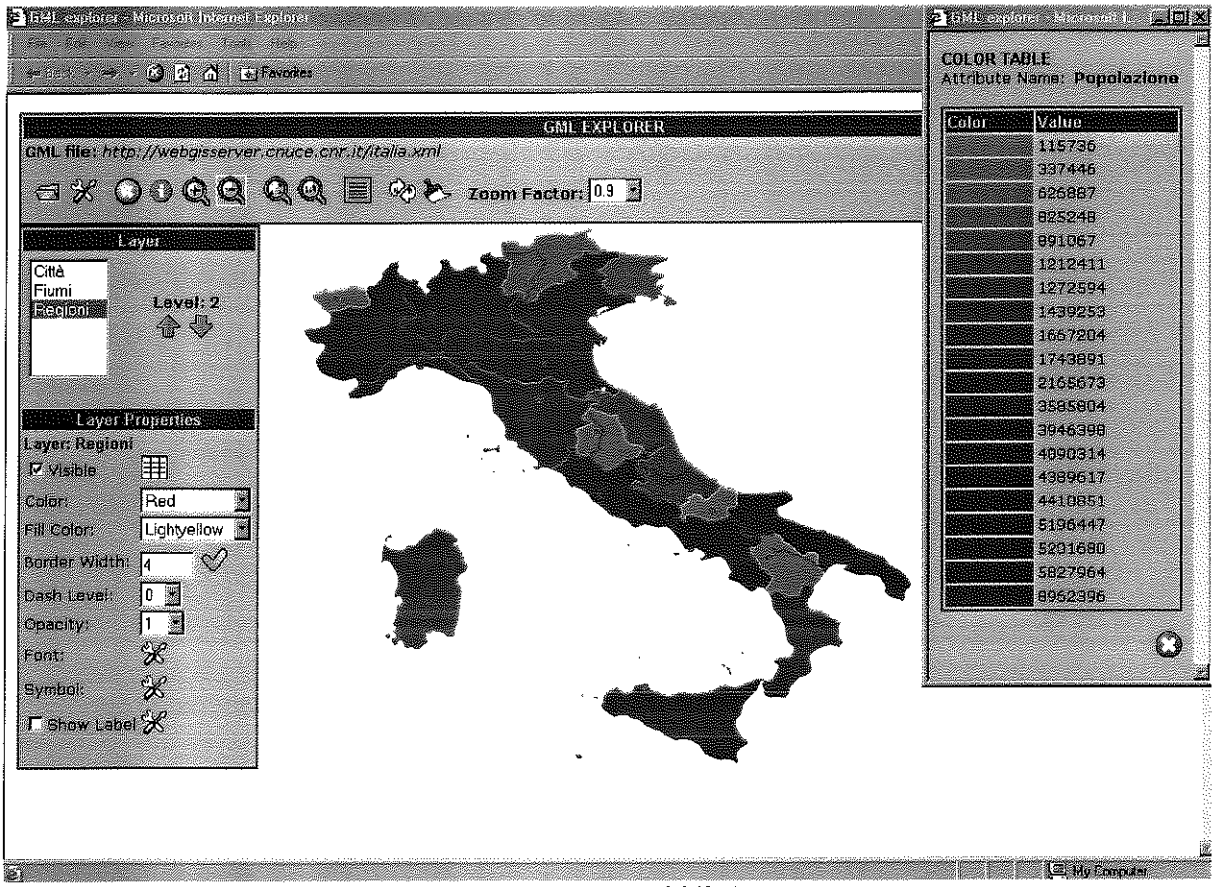


Fig. 5.e – Caricamento del file GML

In questo modo si ottiene la colorazione del documento SVG in funzione del risultato dell'operazione di classificazione (Fig. 5.e).

6. Conclusioni

Il GML è attualmente una tecnologia in fase di evoluzione e questo lavoro si propone di offrire una panoramica su alcuni aspetti dello stato dell'arte, una valutazione dei vantaggi e degli svantaggi connessi. In quest'ottica GML Data Explorer costituisce un'applicazione dimostrativa sull'uso della tecnologia stessa nel settore della costruzione ed interrogazione di mappe tematiche. La codifica dei dati geografici mediante il GML costituisce un approccio che tende a risolvere i sempre più emergenti problemi di interoperabilità legati alla rappresentazione dei dati medesimi.
Il lavoro svolto nel corso della realizzazione di GML Data Explorer, ha permesso di mettere in evidenza alcuni vantaggi e svantaggi connessi all'uso di questa nuova tecnologia.

Vantaggi:

- *Separazione tra contenuto e rappresentazione*: il GML contiene i dati geografici, la loro geometria, le loro caratteristiche ed attributi, ma non indica come questi devono essere rappresentati. La rappresentazione può essere fatta applicando dei fogli di stile in modo da visualizzare i dati in diversi formati grafici (ad esempio SVG, VML, XHTML, etc.) ed

all'interno di un formato personalizzare ulteriormente lo stile (inteso come rappresentazione dei simboli, linee, aree);

- *Interoperabilità*: il GML è un formato testuale standard, non proprietario, nato con l'intento di favorire la condivisione di insiemi di dati eterogenei e di superare le diversità attualmente esistenti tra i vari formati dati GIS. Può essere utilizzato quindi per lo scambio di dati GIS su Internet o come strumento per la conversione di dati di applicazioni GIS differenti.
- *Facilità di manipolazione*: il GML è un'applicazione XML e, come tale, risulta essere facilmente manipolabile attraverso i parser e gli strumenti di sviluppo esistenti per i file XML (DOM API). E' possibile trasformare il contenuto di un file GML in un formato grafico, per la sua visualizzazione, analizzare il contenuto del file per estrarre gli attributi degli elementi geografici o creare altri file GML contenenti solamente l'informazione geografica voluta dall'utente.
- *Indipendenza dall'applicazione*: il GML è indipendente dall'applicazione e non richiede, per il suo utilizzo, alcun software GIS particolare. E' sufficiente avere a disposizione un'applicazione in grado di gestire documenti XML, come ad esempio un web browser. In questo modo si possono realizzare applicazioni, operanti su dati geografici, che non richiedano l'utilizzo, da parte dell'utente, di un particolare applicativo GIS e che siano realizzabili all'interno di un web browser e quindi facilmente utilizzabili anche via Internet (Web-GIS).
- *Elaborazione Client-side*: un file GML contiene le informazioni relative ai dati spaziali e i loro attributi e questo permette di avere a disposizione tutte le informazioni per elaborare i dati lato client. Si può ad esempio visualizzare la mappa od effettuare delle elaborazioni sui dati, lavorando sullo stesso file GML in locale riuscendo a diminuire i tempi di risposta e il carico di lavoro lato server.

Svantaggi:

- *Standard incompleto*: i maggiori problemi di cui risente attualmente il GML sono dovuti principalmente alla mancanza di una specifica completamente definita. La versione attuale (2.0), ad esempio, non risulta completamente definita (Recommendation Paper) per quanto riguarda la specifica del SRS (Spatial Reference System), rendendo inapplicabile il merging di dati con sistemi di riferimento diversi;
- *Gestione attributi*: le specifiche attuali lasciano molta libertà sulle regole di definizione degli attributi non geografici presenti nel documento GML. Infatti è possibile crearsi degli appositi tag per indicare gli attributi utilizzando le regole XML e specificandone la grammatica in appositi file XML Schema o DTD. Se questo può essere inteso come un aspetto positivo per quanto riguarda l'espandibilità del linguaggio, costituisce invece un difficile ostacolo nella realizzazione di applicazioni, non specifiche, che vogliono gestire questi dati.
- *Maggiore interazione SVG-GIS*: il formato grafico maggiormente utilizzato per la rappresentazione degli elementi GML è l'SVG che è un formato nato per la grafica vettoriale sul Web. Questo tipo di grafica ben si adatta alla rappresentazione di dati geografici anche se è carente per quanto riguarda la gestione di alcuni aspetti tipici dei dati geografici. In particolare andrebbe implementato un meccanismo per la gestione dei sistemi di coordinate geografici. Attualmente infatti non esiste alcun meccanismo automatico per la conversione da coordinate geografiche a coordinate di schermo utilizzate dal formato grafico SVG ed è richiesta l'implementazione di funzioni esterne, scritte ad hoc, per risolvere il problema.
- *Incompatibilità tra browser*: i browser attualmente presenti sono in grado di effettuare trasformazioni XSL di documenti XML in HTML o XHTML e di visualizzare l'output senza alcun problema. Purtroppo questo non avviene per formati di output differenti, come ad

esempio SVG, non essendo in grado di richiamare opportunamente il plugin di visualizzazione dopo aver completato la trasformazione del documento. Per risolvere questo problema sono stati utilizzati i Microsoft ActiveXObject per effettuare la trasformazione XSL e per fornire il risultato al plugin SVG. Questo ha permesso di realizzare ugualmente tutte le funzionalità dell'applicazione all'interno del browser imponendo però un vincolo per quanto riguarda il tipo di browser da utilizzare. Infatti gli ActiveXObject sono attualmente supportati solamente dal Microsoft Internet Explorer.

7. Riferimenti

- Lake R. (2001), GML 2.0 – Enabling the Geo-spatial Web, Galdos Systems inc.
- Lake R. (2001), Making maps with Geography Markup Language (GML), Galdos Systems inc.
- Various Authors (2001),Geography Markup Language (GML) 2.0 - OpenGIS® Implementation specification, OpenGIS® Consortium, http://www.opengis.net/gml/01-029/GML2.html
- Cover R. (2001). The XML Cover Page – Geography Markup Language (GML), http://xml.coverpages.org/geographyML.html
- Whiteside A., Bobbitt J., Nicolai R. (2001), Recommended Definition Data for Coordinate References Systems and Coordinate Transformations, OpenGIS® Consortium, http://www.opengis.org/techno/discussions/01-014r5.pdf
- Various Authors (2001), OpenGIS Implementation Specification: Coordinate Transformation Services, OpenGIS® Consortium, http://www.opengis.org/techno/specs/01-009.rtf
- Various Authors (2001), Scalable Vector Graphics (SVG) 1.0 Specification, W3Consortium, http://www.w3.org/TR/2001/REC-SVG-20010904/REC-SVG-20010904.pdf
- Various Authors (2001), XSL Transformations (XSLT) 1.0 Specification, W3Consortium, http://www.w3.org/TR/xslt
- Quin L. (2000), Open Source XML Database Toolkit, Wiley & Sons
- Wilton P. (2000), Javascript guida per lo sviluppatore, Hoepli Informatica
- Elliotte Rusty H. (2001), XML Bible – XLS Transformations, http://www.ibiblio.org/xml/books/bible2/chapters/ch17.html