

LBS avanzati su dispositivi PDA

DI FRANCESCO BARTOLI, FABRIZIO BERNARDINI

II^a parte

La soluzione proposta nella precedente nota prevede tre grandi aree di interesse: il client, il server e il formato dei dati.

Procederemo nel dettaglio considerando tutte le problematiche incontrate nello sviluppo del Geo-Browser e di tutta l'architettura su cui è fondato, che risulta imprescindibilmente indipendente dall'applicazione a partire dal formato di interscambio dei dati geografici.

Il modello GML del formato dati

Un aspetto chiave per la scelta del protocollo e del server è dettato dall'utilizzo di dati in formato vettoriale. La soluzione di usare dati geografici in formato vettoriale significa che gli stessi sono inviati al client in tale formato, invece di usare il formato bitmap, sul quale la maggior parte degli LBS esistenti, che elaborano mappe, si appoggia.

I dati vettoriali possono essere facilmente rappresentati con l'uso dell'XML (*eXtensible Markup Language*).

Una specifica applicazione di XML, chiamata GML (*Geography Markup Language*), è dedicata alla codifica delle entità geografiche, includendo i rispettivi attributi associati ad esse ed è definito dalle specifiche di implementazione rilasciate dall'OpenGIS Consortium (<http://www.opengis.org/docs/>). Il trasferimento di dati geografici descritti in GML può essere facilmente implementato attraverso l'HTTP (*HyperText Transfer Protocol*) che è alla base poi del *World Wide Web*.

GML è un linguaggio di markup usato per codificare informazioni geografiche sia spaziali che non spaziali. Costruito sugli standard ampiamente utilizzati dalla rete Internet e rilasciati dal W3C, GML è usato per esprimere l'informazione geografica in maniera tale che se ne permetta la facile fruibilità attraverso la rete.

Esso non è il primo meta-linguaggio usato per descrivere informazioni geografiche, ma è il primo largamente accettato all'interno della comunità GIS.

Gli altri formati sviluppati prevedono sia la memorizzazione che l'interscambio di informazioni spaziali e temporali, ma supportano tools per lo sviluppo che il più delle volte non sono disponibili perché proprietari.

GML adotta la nozione ISO di *features*: le features descrivono le entità del mondo reale e costituiscono gli oggetti fondamentali in GML. Le features GML possono essere concrete e tangibili o astratte e concettuali, pensando per esempio ai fiumi, edifici, confini, o distribuzioni di quantità su aree geografiche (coperture).

Le features GML sono descritte in termini di proprietà che possono includere descrizioni spaziali, temporali e non spazio-temporali. Per esempio GML può descrivere la posizione, la

forma e l'estensione di oggetti geografici così come proprietà quali il colore, la velocità e la densità con dipendenza anche temporale.

Un disastro naturale come un'inondazione potrebbe essere descritto come una feature dinamica in GML con proprietà di estensione, forma, temperatura dell'acqua, e percentuale di espansione registrate varie volte in certi intervalli di tempo.

Poiché è impossibile descrivere tutte le features e predire il loro uso futuro, il cuore degli *Schemi* GML non contiene la definizione di features concrete, piuttosto queste devono essere definite in *Schemi* di applicazione GML creati dall'utente, quale amministratore del database corrispondente.

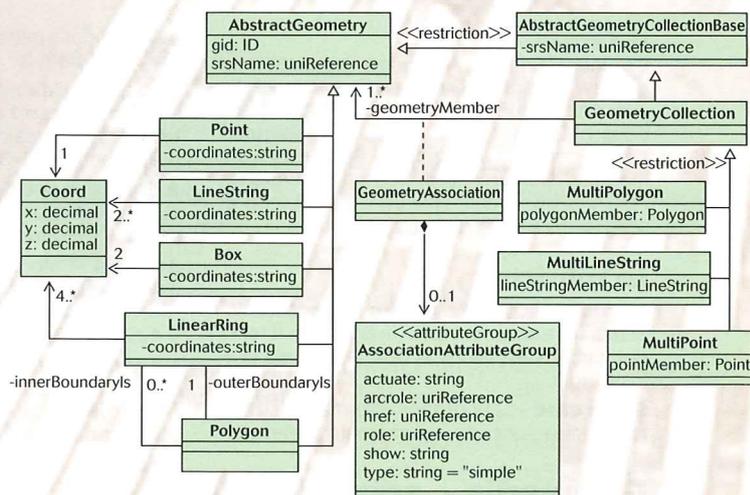


Figura 1: Rappresentazione UML dello Schema *geometry.xsd* del GML

Il nucleo della versione 2.1.1 del GML prevede un insieme di *Schemi* (*feature.xsd*, *geometry.xsd*, *xlinks.xsd*), inclusi in quello creato dall'utente, insieme ad un semplice modello semantico tra oggetti e proprietà che è simile ai diagrammi Relazione-Entità o al modello classe/proprietà.

Usando il modello GML e i suoi componenti, l'utente può descrivere i tipi geografici, che sono usati dentro al proprio dominio di applicazione, cosicché l'insieme di oggetti creati

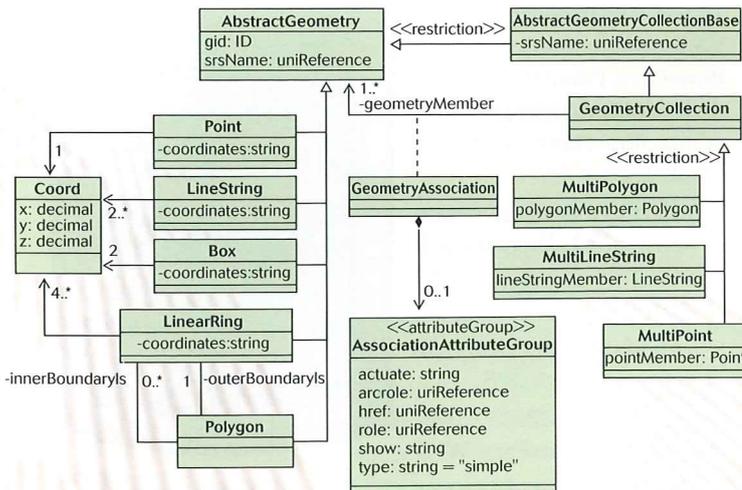


Figura 2: Rappresentazione UML dello Schema feature.xsd del GML

nella forma di uno o più Schemi di applicazione può rispettare il modello semantico e le regole sintattiche del GML.

Per gli elementi ed i tipi geometrici l'utente può decidere di usare quelli già presenti negli Schema feature.xsd e geometry.xsd; la sola cosa che deve sempre fare è definire le proprie features ed i loro tipi. Questo può essere realizzato estendendo i tipi di feature di base descritti nel nucleo degli Schema, e definendo un elemento radice che sia una collezione di features. Tornando al modello delle features nella visione ISO/OGC dell'informazione geografica, si è visto che il termine feature è usato per un oggetto (usualmente) posizionato nello spazio geografico ed anche per modellizzare la rappresentazione dal punto di vista del software.

Una feature avrà un certo numero di proprietà alcune delle quali possono essere geometriche o spaziali. E' necessario evidenziare la seconda: in contrasto con approcci già conosciuti, una feature non è definita primariamente come un oggetto geometrico ma piuttosto come un oggetto concettualmente significativo del quale una o più proprietà possono essere geometriche.

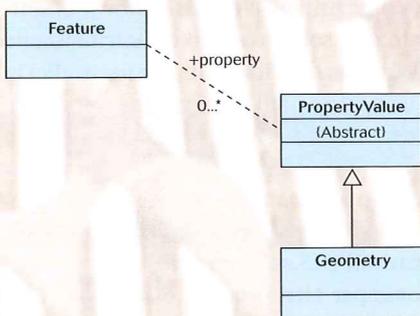


Figura 3: Modello ISO/OGC Feature

L'istanza di una semplice feature in GML potrebbe essere:

```
<abc:Ponte gml:fid = "..">
  <abc:larghezza>40</abc:larghezza>
  <abc:altezza>50</abc:height>
  <abc:tipo>metallico</abc:tipo>
</abc:Ponte>
```

E il corrispondente frammento di Schema di applicazione GML

sarà:

```
<complexType name = "PonteType">
  <complexContent>
    <extension base = "gml:AbstractFeatureType">
      <sequence>
        <element name = "larghezza" type = "integer" />
        <element name = "altezza" type = "integer" />
        <element name = "tipo" type = "string" />
      </sequence>
    </extension>
  </complexContent>
</complexType >
<element name = "Ponte" type = "abc:PonteType"
  substitutionGroup = "gml:_Feature" />
```

Questo dichiara che <abc:Ponte> è un oggetto significativo nel dominio di interesse definito dallo spazio dei nomi riferito dal prefisso "abc".

Uno dei principi basilari del modello concettuale appena esposto è la separazione del contenuto dalla presentazione.

Dal momento che tale linguaggio risulta ancora una novità, non ci sono ancora molti visualizzatori che elaborano dati GML; poiché ci interessa il rendering dei dati provenienti dal lato server per generare una mappa spenderemo qualche parola sui metodi inerenti ad esso.

Sono stati considerati due metodi: uno basato su classi Java che leggono il documento GML prodotto dal server e lo rappresentano tramite API grafiche ed un altro basato sulla trasformazione del GML in SVG (Scalable Vector Graphics), altro formato basato su XML.

Pur essendo SVG una tecnologia ampiamente accettata in ambiente WWW, essa presenta una serie di limitazioni per la nostra applicazione, in particolare i componenti software disponibili risultano solo dei visualizzatori di dati formattati SVG. Quindi per ottenere la massima flessibilità dal Geo-Browser abbiamo deciso di procedere con uno sviluppo software basato sul parsing diretto del GML e sull'uso di un pacchetto di classi Java per il rendering degli oggetti geografici.

Il server WFS

Per la fruizione delle feature geografiche sotto forma di documenti GML l'OGC ha proposto la specifica WFS (Web Feature Server) definendo delle interfacce che permettono operazioni di query e di transazione su feature memorizzate in un database, accessibile attraverso il Web ed organizzato secondo il modello OGC Simple Feature.

Un applicazione client effettua una richiesta di feature ad un

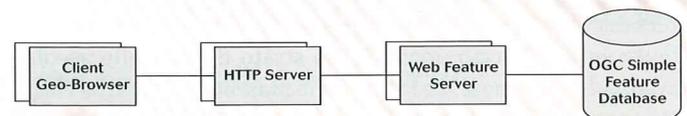


Figura 4: Architettura dell'OGC Web Feature Server

HTTP server, che la inoltra ad un server WFS. Questo legge e processa la richiesta ottenendo le features richieste dal database corrispondente, costruisce il documento GML che le contiene e le invia come risposta al client.

La specifica OGC definisce le seguenti operazioni:

- GetCapabilities: descrive le capacità dello specifico WFS, ossia quali tipi di feature può distribuire e quali operazioni sono ammissibili su ognuna.
- DescribeFeatureType: descrive la struttura di ogni tipo di feature a fronte di una richiesta ricevuta.
- GetFeature: costruisce l'albero GML contenente le feature richieste.
- Transaction: permette di effettuare operazioni di CREATE, DELETE, e UPDATE sulle feature.
- LockFeature: processa una richiesta di inibizione su una o più istanze di un tipo di feature per la durata di una transazione.

Esiste anche un'implementazione open-source di riferimento (GeoServer) dell'OpenGIS Consortium per le specifiche Web Feature Server, usa PostGIS come database di supporto che risulta essere l'estensione spaziale del database PostgreSQL. Nello sviluppo del server abbiamo beneficiato dell'implementazione esistente per l'architettura WFS preoccupandoci solamente di costruire un database di dati GML consistente con il requisito di "leggerezza" del Geo-Browser ed animando gli sforzi nella progettazione e l'implementazione dell'applicazione stessa.

L'implementazione del Geo-Browser

Come anticipato nella nota precedente l'applicazione è stata sviluppata in un linguaggio basato su Java. Il supporto alla tecnologia Java è una importante caratteristica dei moderni dispositivi mobili. Correntemente tutti i terminali Symbian EPOC, i PDA basati su sistema Microsoft Windows e quelli su Palm OS, hanno la possibilità di avere una JVM. Perciò il supporto alla JVM sembra essere una caratteristica standard per i futuri dispositivi di tipo mobile, inclusi quelli di terza generazione.

Il linguaggio Java si presenta per la sua natura flessibile, essendo indipendente dalla piattaforma utilizzata, nello sviluppo di servizi mobili; l'applicazione sviluppata si presenta così abbastanza "leggera" facendo uso di una nuova virtual machine ottimizzata appositamente per terminali PDA e gratuitamente scaricabile (www.superwaba.com.br).

La progettazione dell'architettura software del Geo-Browser è basata sull'analisi effettuata nel lato server e considera i casi d'uso previsti per la fruizione dei servizi offerti dai WFS.

Naturalmente la parte computazionalmente più onerosa è quella del parsing del dato GML e la susseguente visualizzazione tramite le API native messe a disposizione dalla macchina virtuale. Sostanzialmente l'approccio utilizzato per progettare le interfacce utente è basato sul pattern MVC (Model-View-Controller) tramite il quale si separa la visualizzazione dal modello dei dati in maniera tale, che qualsiasi cambiamento del modello si rifletta sulla schermo, mentre uno strato di controllo ascolta eventuali interazioni tra l'utente e le maschere di interfaccia.

Questo ad alto livello è realizzato attraverso due componenti: il parser GML e il Layer Manager. Il cuore della struttura dati è il Layer su cui è costruita la funzione di visualizzazione e che viene aggiornato quando lo strato di controllo lo ritiene necessario. Il parser GML interpreta file GML provenienti da un flusso dati (locale o remoto) in nuovi oggetti Layer che poi vengono aggiunti dentro ad un array di layer.

Al Layer manager è demandato il compito di caricare l'array di layer sullo schermo del dispositivo.

La versione Geo-Browser 0.1 sviluppata in ambiente WindowsCE e PalmOS presenta delle caratteristiche già abbastanza importanti in un ambiente GIS:

- Interfacce grafiche basate sull'uso del dispositivo di input associato al PDA (penna)
- Interfaccia seriale/USB verso ricevitori GPS
- Interfaccia TCP/IP per la connessione ai server WFS
- Composizione di query del tipo <GetFeature>
- Presentazione di un layer georeferenziato contenente le feature ricercate
- Editazione di proprietà delle features
- Editazione di nuove features
- Navigazione delle proprietà delle features contenute nel layer



Figura 5: Test Geo-Browser versione 0.1

Le principali funzioni del Geo-Browser sono accessibili tramite dei piccoli bottoni posti in cima allo schermo:

- P Pan
- Z Zoom
- F Find (una feature, o una proprietà)
- S Show (selezione di features)
- L Layer (switch sui diversi layer)
- N New (features)
- E Edit (caratteristiche delle features)
- M Measure (distanze,...)
- C Configure (mappa)
- ? Info (informazioni sulla feature selezionata)

Conclusioni

E' importante rimarcare come tutti gli elementi hardware, software, protocolli e formato dati sono esistenti e contribuiscono all'implementazione del sistema distribuito di dati geografici in maniera del tutto indipendente dall'applicazione rendendo così l'architettura totalmente generalizzata e implementabile a prescindere dall'uso di dispositivi di tipo PDA.

La capacità di generare mappe dinamiche provenienti da più sorgenti (WFS) rende il Geo-Browser concettualmente minimale e potente in applicazioni GIS sul territorio.

Inoltre si può notare come per altri prodotti o soluzioni GIS, l'applicazione esista anche al di fuori dello stretto contesto geografico non appena il problema incontra la necessità di avere una rappresentazione spaziale (per esempio la descrizione di un sistema di stratificazione del Silicio per un chipset).

Autori

DOTT. ING. FRANCESCO BARTOLI,
Sogesi Spa, Roma: f.bartoli@grupposetec.it

ING. FABRIZIO BERNARDINI,
A&C 2000 Srl, Roma: fb@aec2000.it