

# **Il cGML: una soluzione basata su XML per la rappresentazione di mappe vettoriali su dispositivi mobili**

Roberto Demontis, Emanuela De Vita, Andrea Piras, Stefano Sanna  
{demontis, emy, piras, gerda}@crs4.it

CRS4, Center for Advanced Studies, Research and Development in Sardinia  
Edificio 1, Loc. Piscinamanna, Polaris 09010, Pula (CA), Italia

L'evoluzione tecnologica dei dispositivi mobili, in termini di potenza di calcolo, capacità di visualizzazione e memorizzazione, e il proliferare di sistemi di connessione a banda larga su rete locale *wireless* (WiFi) e cellulare (UMTS), permette la gestione di dati complessi quali le mappe vettoriali e l'accesso ai *Location Based Service* (LBS) per mezzo di un telefono cellulare o un palmare.

L'uso di linguaggi XML nel settore dei LBS condiziona la struttura delle funzionalità e dei dati ma garantisce l'interoperabilità dei servizi. Un esempio è il linguaggio GML (*Geographic Markup Language*) standard de facto nello scambio dei dati tra sistemi GIS e nella rappresentazione dell'informazione geografica. Il GML non è orientato alla rappresentazione di mappe vettoriali, è quindi necessario trasformarlo in un altro linguaggio XML, ad esempio il linguaggio grafico SVG (*Scalable Vector Graphics*).

Per evitare la separazione tra il dato geografico (GML) e la sua visualizzazione (Mobile SVG) ma usando comunque le mappe vettoriali su dispositivi mobili, abbiamo definito il cGML (*compact Geographic Markup Language*). La rappresentazione di un dato tramite cGML è una mappa vettoriale e come tale può essere utilizzata nella presentazione grafica, ma allo stesso momento è anche un contenitore di insiemi di dati geo-riferiti, omogenei per semantica, utilizzabili in elaborazioni locali sul dispositivo mobile.

*Enhanced mobile device performances, in terms of CPU power, memory capacity, display capabilities and adoption of wideband connectivity based on local wireless (WiFi) and cellular (UMTS) networks, bring interactive mapping and access to Location Based Services (LBS) to mobile phones and PDAs. The use of XML language in the LBS influences data structure, functionalities and performances; although it could be time and memory expensive, it guarantees service interoperability. The GML language (Geographic Markup Language) is "de facto" standard in data exchanging between Geographic Information Systems. The GML has not been designed to represent vector map: it has to be transformed into another XML language, for example the graphical-oriented XML language Mobile SVG.*

*To avoid the separation between the geographic data (GML) and its visualization (Mobile SVG), keeping vector mapping on mobile devices, we have designed the cGML (compact Geographic Markup Language). The data encoded using cGML is a vector map, so that it can be used in the graphic representation, but at the same time it is a container of geographical related data set, homogenous for semantic and usable in local processing on the mobile device.*

## **Introduzione**

La ricerca e lo sviluppo di applicazioni per dispositivi mobili è in continua evoluzione, in particolare per ciò che attiene alla localizzazione e all'accesso a dati complessi geo-riferiti. In generale l'uso di basi di dati geo-referenziate è necessario quando il servizio offerto è legato alla posizione dell'utente. In questo contesto è usuale utilizzare mappe quale rappresentazione del dato geografico e/o dinamica interattiva nella fruizione di informazioni geo-riferite.

Lo stato dell'arte annovera diverse soluzioni sia nella progettazione di servizi che utilizzano la posizione dell'utente quale chiave di accesso a dati strutturati, i *Location Based Services* (LBS), sia nelle applicazioni di semplice consultazione cartografica.

Le prime applicazioni cartografiche su dispositivi mobili utilizzano formati proprietari per l'archiviazione, la trasmissione e la gestione dell'informazione. In una prima fase i dati relativi ad un'area d'interesse (ad esempio, una città o un quartiere) sono pre-caricati sul dispositivo. In alcuni casi, quando l'utente consulta le informazioni, queste sono integrate con dati *real-time* ricevuti da un *server* remoto.

Le architetture più recenti tendono invece a utilizzare tecniche in cui vi è un continuo scambio di informazioni tra il *server*, che genera i dati, ed il *client*, che li richiede. Ciò è dovuto alla quantità e alla qualità dell'informazione offerta. Non a caso attualmente si pone l'accento su una strutturazione su più livelli dell'informazione, basata sulle differenti fasi di consultazione e sulla definizione del dettaglio dei contenuti. In tali architetture, i diversi componenti dialogano utilizzando linguaggi basati su XML, la cui flessibilità rende possibili soluzioni scalabili orientate alla definizione ed all'uso di standard aperti. Nonostante le limitazioni dei dispositivi mobili, è possibile trovare numerose analogie con le applicazioni che gestiscono la pubblicazione di dati geografici tramite Internet, le applicazioni Internet basate su sistemi GIS.

## **Limiti nei dispositivi mobili**

Durante lo sviluppo di applicazioni per dispositivi mobili è necessario tener conto delle limitate caratteristiche della piattaforma *target*. In particolare, capacità di calcolo, memoria volatile e persistente e le limitate dimensioni dei *display* impongono approcci ad hoc sia a livello di progettazione che di implementazione. Inoltre, dispositivi appartenenti ad una medesima fascia di mercato possono avere caratteristiche tecniche completamente differenti e imporre modifiche sostanziali al codice dell'applicazione.

Altro aspetto critico dell'uso di dispositivi mobili è legato all'infrastruttura di comunicazione, generalmente basata su reti di telefonia cellulare a banda stretta. Se l'ampiezza di banda è destinata ad aumentare, con l'avvento delle tecnologie EDGE e UMTS, rimangono i problemi legati alla mobilità degli utenti e alla difficoltà di copertura in numerose aree extraurbane. Per cui l'informazione destinata al dispositivo mobile deve essere compatta per un rapido trasferimento e, allo stesso tempo, deve garantire un certo grado di atomicità che ne permetta un utilizzo, anche parziale, in caso di interruzione della connessione. Queste problematiche, unite a requisiti di ergonomia, usabilità e ottimizzazione del consumo energetico del dispositivo, costituiscono vincoli primari nella progettazione delle applicazioni.

Tra le diverse soluzioni presenti sul mercato, il profilo *Mobile Information Device Profile* (MIDP) di *Java2 MicroEdition* (J2ME), costituisce una piattaforma flessibile e multidispositivo per l'implementazione e la distribuzione di applicazioni per terminali cellulari. J2ME è basata su un sottoinsieme del linguaggio Java e ha un'architettura costituita da tre livelli modulari (VM, configurazione e profilo) che consente di scrivere applicazioni in grado di adattarsi ai limiti tecnologici sopraesposti e trarre vantaggio dal vasto numero di soluzioni disponibili per la piattaforma *desktop*.

## **GML e interoperabilità**

Lo standard de facto utilizzato nella rappresentazione del dato geografico è il GML (*Geography Markup Language*). Questo linguaggio è nato proprio dall'esigenza di avere strumenti adeguati per garantire l'interoperabilità tra sistemi GIS.

In questo contesto si può considerare che il dato geografico di base utile alla generazione di una mappa vettoriale in un generico LBS sia nel formato GML. Vi sono alcune controindicazioni nell'uso diretto del GML sui dispositivi mobili: un documento in formato GML può essere di dimensioni tali da richiedere lunghi tempi per il trasferimento attraverso reti di telefonia. Inoltre, le operazioni di codifica e decodifica dei dati nel formato GML, così come il salvataggio in locale, richiedono un considerevole spazio di memoria. Eventuali tecniche basate su compressione e decompressione del dato richiederebbero ulteriori risorse in termini di memoria ed elaborazione. Per questo motivo, si ricorre alla trasformazione in linguaggi più snelli e specializzati nella rappresentazione grafica dei dati (Reichenbacher T., 2003). Varie soluzioni legate ad Internet GIS propongono l'uso del profilo mobile di SVG (*Scalable Vector Graphics*) quale linguaggio utile a rappresentare mappe vettoriali.

Mobile SVG è adatto alla sola codifica delle operazioni di visualizzazione escludendo fasi intermedie di elaborazioni del dato. Ad esempio, per aggiungere informazioni di carattere geografico, quali il Sistema di Riferimento Spaziale (SRS) usato, sono necessari elementi esterni a SVG.

Per semplicità d'implementazione, il lato *client* è un browser specifico per il linguaggio grafico del servizio ma in questo modo si determina un impoverimento della dinamica interattiva tra servizio ed utente, tendendo a far aumentare il numero di richieste (necessarie per effettuare, ad esempio, interrogazioni sui dati visualizzati) e quindi il tempo di latenza dovuto alle trasmissioni. Possiamo definire questa soluzione come *mobile browsing*. L'aumentare delle prestazioni dei dispositivi mobili conducono alla definizione del paradigma del *mobile computing*, in cui il terminale non è solo strumento di visualizzazione ma fornisce funzioni di elaborazione locale, gestione dei dati e supporto sul campo in funzione della posizione e del profilo dell'utente. Nasce dunque l'esigenza di avere uno strumento in grado di fornire le caratteristiche essenziali dei linguaggi XML ma, al tempo stesso, ne permetta un uso efficace ed efficiente su dispositivo mobile.

## **Compact Geographic Markup Language, cGML**

Basandoci sulle considerazioni espresse nei paragrafi precedenti, abbiamo definito il cGML (*compact Geographic Markup Language*): un linguaggio XML che permette di unire informazioni geografiche e di *rendering* in un unico documento e di eseguire elaborazioni lato *client* (Piras A. et al., 2004).

cGML è definito da tre XML-Schema: uno specifica gli elementi geometrici, un altro gli elementi geografici e il terzo definisce la struttura di un documento cGML.

La corrente versione di cGML deriva da GML 2.1.2 (Lake R. et al., 2002), alcune sue caratteristiche sono:

- I nomi degli elementi cGML sono più corti rispetto ai loro corrispondenti nel linguaggio GML, seppur mantenendo la caratteristica di leggibilità.
- Non sono stati definiti elementi privi di attributi aventi il solo scopo di contenere altri elementi (come `featureMember` di GML).
- Numero ridotto di attributi disponibili.
- Nessun supporto per *XLink*.

In questo modo, cGML permette di codificare l'informazione geografica in modo più compatto di quanto sia possibile in GML, da ciò la parola "*compact*" presente nella sua definizione. Con la definizione dell'elemento *root* cGML è possibile scrivere un documento nel linguaggio cGML come un documento XML valido, cosa non realizzabile con GML. Infatti, sebbene GML definisca molti

elementi non ha un elemento *root* e i suoi elementi devono essere utilizzati all'interno di altri documenti XML.

Il primo elemento figlio dell'elemento *root* è *Head*, che identifica il Sistema di Riferimento Spaziale (SRS), l'area geografica d'interesse in cui gli elementi geografici sono contenuti e l'area di visualizzazione sul dispositivo.

L'indicazione del SRS fornisce le informazioni per la trasformazione delle coordinate da reali a cartografiche ed è contenuto nell'attributo *srsName* nell'elemento *RealBox*. Nel GML tale indicazione può essere specificata per ogni elemento geografico mentre in cGML è permesso definire un unico SRS valido per tutti gli elementi geografici. I valori contenuti nell'elemento *RealBox* sono le coordinate reali dell'area d'interesse (AOI).

A *RealBox* segue l'elemento *View*, i cui valori interi sono le dimensioni in *pixel* della vista sul dispositivo. L'attributo *zoom* di *View* è il fattore di scala tra l'area d'interesse e la vista e, più precisamente, è l'inverso del numero di unità reali associate ad un *pixel*.

In *Esempio 1*, l'elemento *Head* è contenuto nelle righe dalla 3 alla 8 e `EPSG:32632` identifica il SRS UTM (WGS84) Zone 32 Nord.

```

1: <?xml version="1.0" encoding="UTF-8"?>
2: <cGML xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="cgml_base.xsd" version="1.0">
3:   <Head>
4:     <RealBox srsName="EPSG:32632">
5:       <cds>510775,4339616 512275,4341116</cds>
6:     </RealBox>
7:     <View zoom="0.2667">400,400</View>
8:   </Head>
9:   <FtCl id="Shapel" name="aree">
10:    <Ft id="a1" name="mainstreet">
11:      <LnSt>
12:        <cds>267,39 276,42</cds>
13:      </LnSt>
14:      <info>info della feature a1 in Shapel</info>
15:    </Ft>
16:    <Ft id="b6">
17:      <LnSt>
18:        <cds>326,64 353,78 396,102</cds>
19:      </LnSt>
20:      <info>info della feature b6 in Shapel</info>
21:    </Ft>
22:  </FtCl>
23:  <Ft id="p_o" name="ufficio postale">
24:    <Point>
25:      <cds>79,10</cds>
26:    </Point>
27:  </Ft>
28: </cGML>

```

### *Esempio 1. Un documento cGML.*

Definendo l'area di vista come  $V(0, 0, larghezza, altezza)$ , la AOI come  $A(bx1, by1, bx2, by2)$ , le coordinate reali  $P_{real}(x, y)$  corrispondenti ad un punto della vista  $P_{view}(x_v, y_v)$ , si ottengono applicando le regole di trasformazione [1] e [2].

$$x = \text{round}(\text{zoom} * x_v) + bx1 \quad [1]$$

$$y = \text{round}(\text{zoom} * (\text{height} - y_v) + by1 \quad [2]$$

$$x = \text{round}(0.2667 * 25) + 510775 = 510782 \quad [3]$$

$$y = \text{round}(0.2667 * (400 - 187) + 4339616 = 4339673 \quad [3]$$

Utilizzando i valori in *Esempio 1* (un valore di *zoom* pari a 0.2667, che corrisponde ad un fattore di scala di una vista che prevede 3.75 metri per *pixel*, e un'area di vista  $V(0, 0, 400, 400)$ ), al punto della vista  $P_{view}(25, 187)$  corrisponde il punto  $P_{real}(510782, 4339673)$  in coordinate reali. L'elemento *Head* è seguito da una lista di singoli elementi geografici (gli elementi *Ft*) o da una lista di loro insiemi (gli elementi *FtCl*). Tali elementi sono definiti basandosi sugli elementi

astratti `_Feature` e `_FeatureCollection` del GML e con l'intento di definire degli elementi concreti e più semplici, infatti: l'elemento `featureMember` del GML viene rimosso, l'attributo opzionale `id` di `_Feature` e `_FeatureCollection` è necessario in `Ft` e `FtCl`, l'attributo `name` diviene opzionale in `cGML` e l'elemento interno `description` di GML è ridenominato in `info`. In *Esempio 1* vi è una collezione di elementi con `id` uguale a "Shape1" (righe 9 – 22) contenente due elementi `Ft` (righe 10 – 15 e 16 – 21), seguita da un elemento `Ft` (righe 23 – 27).

All'interno degli elementi `Ft` si trovano gli elementi geometrici del dato geografico ovvero la topologia. Gli elementi delle primitive geometriche del `cGML` sono pressoché gli stessi del GML 2.1.2, sebbene il nome sia più breve. Le differenze tra tali primitive riguardano l'introduzione degli elementi `cGML Arc`, `MlArc` (*multi-arc*) e `MlLnRn` (*multi-linear ring*) in `cGML` e l'esclusione dell'elemento `Box` di GML. Gli altri elementi geometrici del `cGML` sono: `Point`, `LnSt` (*line string*), `LnRn` (*linear ring*), `Plgn` (*polygon*), `MlPoint` (*multi-point*), `MlLnSt` (*multi-line string*), `MlPlgn` (*multi-polygon*) e `MlGeo` (*multi-geometry*).

Ogni elemento geometrico, al suo interno definisce i limiti della figura come sequenza di coordinate dei suoi vertici o coordinate del centro, raggio e angolo d'inizio e fine nel caso di `Arc`. Le coordinate sono riferite ad un SRS planare ed è usata interpolazione lineare tra due vertici.

In GML, le coordinate dei vertici delle geometrie si possono rappresentare attraverso una sequenza di elementi `coord` (i quali incapsulano gli elementi `X`, `Y` e `Z`) oppure come una singola stringa contenuta all'interno dell'elemento `coordinates`. In `cGML` è stato scelto questo secondo tipo di codifica perché permette di ridurre il numero di caratteri nell'esprimere la stessa lista di coordinate.

Il nome `coordinates` è modificato nel più breve `cds` ed il suo valore è una sequenza di gruppi di numeri separati da uno o più spazi, con ogni gruppo composto da uno o più valori interi separati da una virgola. Le coordinate sono riferite alla vista sul dispositivo mobile. Le righe 5, 12, 18, 25 di *Esempio 1* mostrano esempi di `cds`.

### Un caso d'uso

Di seguito descriviamo un caso d'uso che prevede i telefoni cellulari quali dispositivi mobili. Lo scenario si riferisce alla ricerca di una camera d'albergo da parte di un turista e per ogni albergo deve essere specificato il costo e il numero delle camere disponibili. Il turista ha a disposizione sul dispositivo mobile un'applicazione con cui inserisce le informazioni per la ricerca. Il *client* si connette al *server* attraverso il protocollo HTTP e invia una richiesta che specifica la AOI e il *layer* per il quale vuole massimo dettaglio d'informazioni. Il *server* analizza la richiesta e risponde inviando un documento `cGML` generato utilizzando le tipologie e le sorgenti dei dati individuate nella fase iniziale della progettazione. Tali informazioni sono state archiviate nel formato `cGML` unitamente all'implementazione dei meccanismi di reperimento dei dati dinamici, ottenuti tramite interrogazioni SQL ad un *database* relazionale. Tutte le scelte adottate servono a ridurre la dimensione dei dati e garantire un `cGML` di grandezza contenuta con scala di acquisizione riferibile alla scala di rappresentazione finale su dispositivo.

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
<RealBox srsName="EPSG:32632">
  <cds>510505,4339272 512871,4341638</cds>
</RealBox>
<View>400,400</View> <Layer id="hotel"/>
</Request>
```

#### *Esempio 2. Richiesta dell'applicazione client.*

L'applicazione *client* elabora il documento e l'utente può decidere se memorizzarlo o meno in maniera persistente prima di visualizzarlo. E' possibile anche decidere quali e in quale ordine far disegnare i diversi *layer* sul *display*, e chiaramente fare operazioni di *zooming*, *panning*, ricerca per

nome di un punto di interesse e calcolo e visualizzazione del percorso minimo tra due punti. In questo modo si può utilizzare in maniera autonoma la parte *client* senza stabilire una nuova connessione col *server*, evidenziando così le potenzialità di un'applicazione basata sul paradigma del *mobile computing*. Ciò è attuabile basandosi sul modello dei dati espresso dal cGML in quanto si possono identificare univocamente l'insieme di elementi geografici. Ad esempio è possibile identificare le strade, la loro topologia e poter quindi calcolare percorsi minimi. Ogni informazione relativa allo stile di rappresentazione è parte integrante del servizio e non presente nel dato in formato cGML.

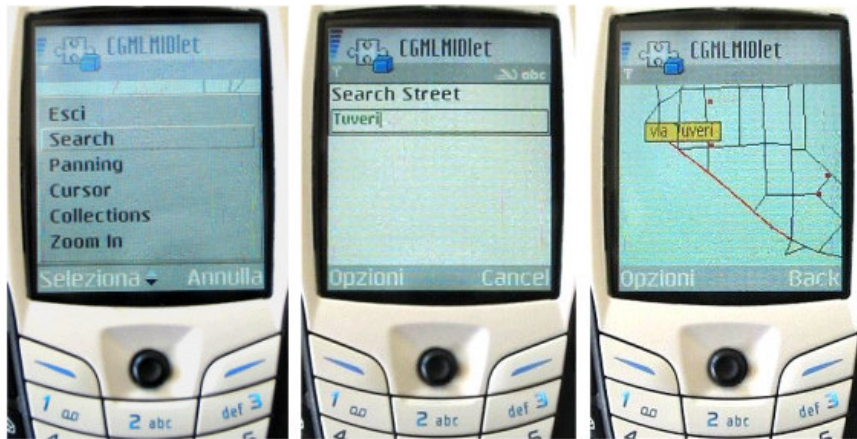


Figura 1. Esempi di utilizzo dei dati rappresentati con il linguaggio cGML.

## Conclusioni

La gestione di LBS con moduli di calcolo su dispositivi mobili richiede la definizione di nuovi linguaggi compatti, utili sia per la visualizzazione di mappe vettoriali, ma anche quali contenitori di insiemi di dati geo-riferiti omogenei per semantica e relativi ad una specifica area d'interesse, rendendo possibile l'uso del modello dei dati rappresentato per elaborazioni locali sul dispositivo. Il cGML soddisfa questi requisiti e si è dimostrato una soluzione utilizzabile su dispositivi con forti limitazioni in termini di potenza di calcolo, memorizzazione e visualizzazione.

L'introduzione della notazione compatta e l'eliminazione di parte degli elementi GML determinano vantaggi nella manipolazione dei dati nei dispositivi mobili. I documenti cGML sono in media più corti del 64% dei documenti GML che contengono gli stessi dati geografici e quindi:

- Il trasferimento del documento in reti *wireless* richiede meno tempo ed è più economico.
- Non è richiesta compressione lato *server* e decompressione lato *client*.
- La diminuzione della profondità dell'albero Document Object Model (DOM) permette ai parser XML di utilizzare una quantità di memoria inferiore per le elaborazioni.

L'articolo inoltre ha l'intento di evidenziare la necessità di standard per la gestione del dato nelle applicazioni per dispositivi mobili, cioè standard specifici nella rappresentazione del dato geografico piuttosto che della sua sola rappresentazione grafica.

## Riferimenti

- Piras A. et al., (2004), Compact GML: Merging Mobile Computing and Mobile Cartography, *In Proceedings of the GML And Geo-Spatial Web Services Conference 2004*,  
 Reichenbacher T., (2003), Adaptive methods for mobile cartography, *In Proceedings of the 21st International Cartographic Conference (ICC) 2003, ISBN: 0-958-46093-0, 1311-1320*  
 Lake R. et al., (2002), OpenGIS® Geography Markup Language (GML) Implementation Specification, version 2.1.2., <http://www.opengis.org/docs/02-069.pdf>