

GeoServer, il server open source per la gestione interoperabile dei dati geospaziali

Ing. Simone Giannecchini, Ing. Andrea Aime

GeoSolutions S.A.S, Via Carignoni 51, 55041 Camaiore (LU), Italia
Tel. +390584962313, Fax. +390584962313, email: info@geo-solutions.it

Abstract

GeoServer è un server GeoSpaziale Open Source scritto in Java, seguendo le comuni pratiche Java Enterprise, per la gestione, disseminazione e analisi di dati geospaziali.

GeoServer permette di distribuire, manipolare e analizzare i dati usando i più diffusi standard OGC (WMS, WFS, WCS, WPS), senza però dimenticare estensioni specifiche per una interazione trasparente con client come Google Earth e software commerciale in genere, né gli ormai diffusi approcci basati su protocolli REST e GeoJSON per la distribuzione semplificata di semplici dati vettoriali. Obiettivo della presentazione è fornire allo spettatore una esauritiva panoramica delle funzionalità offerte da GeoServer per la creazione di Spatial Data Infrastructure interoperabili, con particolare attenzione alle nuove funzionalità introdotte in GeoServer 2.1 e all'analisi di dati spaziali con il protocollo WPS 1.0. *GeoServer is an open source geo-spatial server written in Java, following the common Java Enterprise practices, allowing for the handling, distribution and analysis of geospatial data. GeoServer allows to distribute, handle and analyses data using the most widely accepted OGC standards (WMS, WFS, WCS and WPS), without forgetting specific extensions for a transparent interacting with clients such as Google Earth and commercial software in general, and providing support for the now common protocols based on REST and GeoJSON for the distribution of simple vector based data. The presentation will give the audience an exhaustive overview of GeoServer functionalities for the creation of interoperable Spatial Data Infrastructures, with particular focus on the new GeoServer 2.1 functionalities and the WPS 1.0 spatial data analysis capabilities.*

Introduzione

GeoServer¹ è un server open source rilasciato sotto licenza GPL (Free Software Foundation, 1991) per la gestione e la disseminazione di dati geospaziali secondo specifiche edite dall'*Open Geospatial Consortium (OGC)*² e dall'*ISO Technical Committee 211 (ISO TC 211)*, di conseguenza fornisce le funzionalità di base per creare Infrastrutture di Dati Territoriali, dette anche *Spatial Data Infrastructure (SDI)*, basate su tali standard. GeoServer è stato sviluppato per ingerire, gestire e servire dati geospaziali sia vettoriali (*feature* nel gergo OGC) che raster (*gridcoverage* nel gergo OGC) ma anche per creare e disseminare mappe georeferenziate ottenute giustapponendo versioni renderizzate dei dati menzionati in precedenza secondo specifiche regole di stile, codificate secondo gli standard menzionati. Obiettivo di questo articolo è fornire una introduzione ragionata a quello che sono le caratteristiche e le funzionalità offerte dal GeoServer ponendo l'accento soprattutto sul supporto per la interoperabilità tramite la implementazione di protocolli standard. Inoltre saranno fornite anche informazioni rispetto alla architettura interna del GeoServer nonché una serie di informazioni relative al progetto Open Source stesso che lo gestisce e promuove.

¹ <http://geoserver.org/>

² <http://www.opengeospatial.org/>

Caratteristiche Salienti

Nelle successive sezioni introdurremo e descriveremo quelle che sono le caratteristiche salienti del GeoServer coprendo con adeguato dettaglio protocolli supportati, funzionalità offerte, formati di dati supportati in ingresso ed in uscita nonché la infrastruttura interna.

Protocolli Supportati

GeoServer, come accennato in precedenza, è stato creato per fornire il supporto a protocolli standard in modo da rappresentare un valido strumento per la creazione di infrastrutture distribuite geospaziali interoperabili. Esso supporta in modo nativo una vasta gamma di standard, sia promulgati da enti internazionalmente riconosciuti come OGC ed ISO TC 211, sia standard de facto creati all'interno della comunità degli sviluppatori di software geospaziale Open Source, quali ad esempio *WMS-C*, il protocollo l'accesso orientato ai *tile* del *Web Map Service* o *WMS*.

Gli standard attualmente supportati sono:

- **OGC Web Map Service (WMS) 1.1.1** (de La Beaujardière, et al., 2002) con supporto per **OGC Styled Layer Descriptor (SLD) 1.0.0** (Lalonde, 2002), **OGC Web Map Service (WMS) 1.3** (de la Beaujardiere, 2006) con supporto per **OGC Symbology Encoding (SE) 1.1** (Muller, 2006) per la generazione e disseminazione di mappe georeferenziate a partire da dati sia raster che vettoriali. GeoServer supporta inoltre tutta una serie di direttive di rendering proprietarie, alcune delle quali saranno discusse nel seguito
- **OGC Web Coverage Service (WCS) (Whiteside, et al., 2006) 1.1.0** and **Web Coverage Service (WCS) (Evans, 2003) 1.0.0** per la gestione e disseminazione dei dati raster in formato nativo. In altre parole questo servizio fornisce la possibilità di accedere a sottoaree, riproiettare, ricampionare e cambiare formato dei dati raster. Notare che il WCS differisce dal WMS in quanto nella catena di processamento del dato raster non vi sono direttive di rendering
- **OGC Web Feature Service (WFS) 1.0.0** (Vretanos, 2002) e **1.1.0** (Vretanos, 2005). Questo tipo di servizi permette la gestione e la disseminazione di dati vettoriali in formato nativo; è evidente il parallelo con il servizio WCS per i dati raster: tramite WFS è possibile avere accesso diretto al dato vettoriale originale, o comunque ad una versione riprocessata di esso secondo le indicazioni ricevute, piuttosto che ad un rendering degli stessi come tramite attraverso il servizio WMS. E' opportuno sottolineare che GeoServer supporta come formato di output il **Geography Markup Language (GML)** insieme a altri formati di largo uso quali Shapefile (ESRI, 1998) e GeoJSON
- Eccellente supporto **OGC Keyhole Markup Language (KML)** (Wilson, 2008) il dialetto XML che permette al GeoServer di interagire con Google Earth e Google Maps anche con caratteristiche avanzate che analizzeremo nel prosieguo
- **WMS-C**, **OGC WMS-T**, **TMS** attraverso la sua estensione GeoWebCache³
- **OGC Web Processing Service (WPS) 1.0.0** (Schut, 2007) in modo da fornire supporto per la pubblicazione interoperabile di geoprocessi verso il web
- **GeoRSS** (Geo), lo standard de facto per estendere il formato di trasporto Remote Syndication Standard (RSS) in modo da aggiungere supporto per specificare topologie georiferite

Nella tabella che segue sono riassunte le versioni dei protocolli OGC che il GeoServer implementa, con particolare riferimento a quelli per cui è *Reference Implementation*⁴.

³ <http://geowebcache.org/>

⁴ Una *Reference Implementation* è una implementazione di una specifica OGC che è stata verificata e validata come pienamente corretta e rispondente alla specifica e che solitamente viene messa a diposizione usando software Open Source e gratuito in modo che altre implementazioni anche proprietarie possano usarla come riferimento. Si veda questa pagina per ulteriori informazioni <http://cite.opengeospatial.org/reference>.

Protocollo	Versione	Reference Implementation
WCS	1.0	
	1.1.0	X
WFS	1.0	X
	1.1	X
WMS	2.0 ⁵	
	1.1.1	
	1.3	
WPS	1.0.0	

Figura 1 – Protocolli standard supportati dal GeoServer con le relative versioni.

Formati Supportati

La lista di formati supportati in ingresso ed in uscita dal GeoServer è piuttosto vasta, grazie anche ad un largo numero di estensioni a disposizione che contribuiscono ad incrementarne l'estensione, mentre nuovi formati sono aggiunti continuamente dai membri della comunità degli sviluppatori.

Formato	WMS		WFS		WCS	
	Input	Output	Input	Output	Input	Output
Geotiff	I	I	-	-	I	I
Image+world file	I	I	-	-	I	I
GTOPO30	I	I	-	-	I	I
Ascii grid	I	-	-	-	I	I
Image mosaic	I	-	-	-	I	-
Image pyramid	I	-	-	-	I	-
ArcSDE raster	I*	-	-	-	I*	-
Erdas Image	I	NP	-	-	T	-
ECW	I	NP	-	-	T	-
MrSID	I	NP	-	-	T	-
JPEG2000	I	NP	-	-	T	-
DTED	I	NP	-	-	T	-
NITF	I	NP	-	-	T	-
CADRG	I	NP	-	-	T	-
Grass Binary Grid	D	-	-	-	D	
netCDF-CF	D	-	-	-	D	P
GriB1	D	-	-	-	D	P
HDF-EOS	D	-	-	-	D	P
Oracle GeoRaster	I*	-	-	-	I*	-
BigTiff	I	-	-	-	I	I*
ShapeFile	I	-	I	I	-	-
Postgis	I	-	I	-	-	-
ArcSDE feature	I	-	I	-	-	-
Oracle	I	-	I	-	-	-
MySQL	I*	-	I*	-	-	-
SQL server	I*	-	I*	-	-	-
OGR supported	D	-	I*	I	-	-

⁵ In fase di sviluppo.

formats						
Cascading WFS	I*	-	I*	-	-	-
Cascading WMS	I*	-	-	-	-	-
SpatialLite	I*	-	I*	-	-	-
Teradata	I*	-	I*	-	-	-

Figura 1 – Formati dati supportati da GeoServer (I = implementato, I*=implementato ma non maturo, T= in fase di test, D= in fase di sviluppo, P=pianificato, NP= non pianificato).

Infrastruttura Enterprise

GeoServer è sviluppato basandosi sul framework Spring⁶, che attualmente è il framework Java Enterprise più utilizzato in ambiente web. E' fuori dagli scopi di questo articolo introdurre e discutere tale framework, nonostante ciò riteniamo sia importante rimarcare alcune caratteristiche fondamentali del GeoServer e che esso espone facendo leva sulle potenzialità di Spring:

- **Modularità:** GeoServer è costituito da un vasto numero di moduli, sia *core*, moduli necessari al corretto funzionamento di base della piattaforma, che di *estensione*, moduli che aggiungono nuove funzionalità alla piattaforma oltre a quelli di base, sia *verticali*, ossia che forniscono servizi all'utente finale, che *orizzontali*, ossia che aggiungono funzionalità a livello di piattaforma piuttosto che di servizi verso l'utenza. Ad esempio, uno dei moduli *core* del GeoServer è il modulo che gestisce la configurazione interna del server mentre il modulo *WPS* è una estensione che fornisce supporto al protocollo WPS appunto. Nella fattispecie, il primo modulo è anche *orizzontale* mentre il secondo è *verticale*. Inoltre questa modularità ha permesso di consentire una elevata indipendenza fra i vari moduli funzionali aspetto fondamentale per un framework di queste dimensioni
- **Estendibilità:** il GeoServer mette a disposizione chi volesse aggiungere nuove funzionalità una gamma vastissima di *extension points* sia sotto forma di interfacce Java da implementare sia sotto forma di punti di estensione messi a disposizione usando i cosiddetti *Bean* di Spring
- **Configurabilità:** il GeoServer è completamente ed estensivamente configurabile tramite interfaccia grafica, tuttavia è opportuno sottolineare come tutti i moduli, seguendo gli standard di Spring, siano ampiamente configurabili via file XML in modo da creare versioni altamente personalizzate del server
- **Architettura Multilivello:** nel GeoServer è stato mantenuto un alto grado di separazione fra i vari livelli della infrastruttura, *dati, logica di servizio e interfaccia*

Come illustrato in **Figura 2** il GeoServer, si basa, oltre che su Spring, su tutta una serie di librerie Open Source molto conosciute, tra cui vorremmo citare:

- **GeoTools (Geo11):** toolkit Open Source geospaziale scritto in Java che permette la gestione, manipolazione e rendering di dati sia raster che vettoriali. Esso costituisce la base operativa con cui GeoServer si interfaccia e gestisce i dati geospaziali.
- **ImageIO (Oracle, 2011), ImageIO-Ext (GeoSolutions, 2007):** librerie di input/output dei dati geospaziali
- **Java Topology Suite (JTS) (Davis, 2002):** libreria Open Source geospaziale scritta in Java per la gestione topologica delle geometrie
- **GDAL:** libreria Open Source per l'accesso ai dati raster geospaziali
- **Java Advanced (JAI) (Oracle, 2001), JAITools (Bedward, 2009):** librerie Open Source in Java per il processamento ad alte prestazioni dei dati raster

⁶ <http://www.springsource.org/>

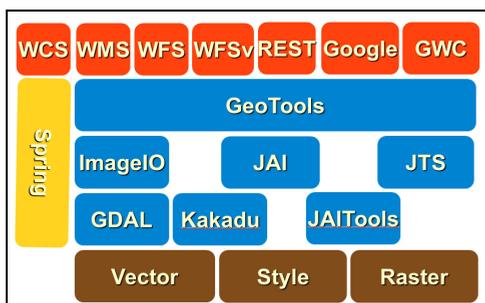


Figura 2 – Infrastruttura di base del GeoServer.

GeoServer è stato utilizzato con successo con una vasta gamma di application server come Tomcat, Jetty, WebLogic, JBoss, etc., inoltre, basandosi su Spring, è stato sviluppato in modo da poter utilizzare risorse *managed* a livello server come *connection pool* di database.

Interfaccia di Amministrazione Web

Una delle caratteristiche più apprezzate del GeoServer è il fatto che esso fornisce una interfaccia di amministrazione completamente accessibile via web che permette di gestire ogni aspetto della configurazione sia dei servizi che dei dati. E' cross-browser e non richiede installazioni di alcun plug-in (si veda **Figura 3**).

In particolare tramite questa interfaccia sono possibili le seguenti azioni:

- Aggiunta e configurazione di nuove fonti di dati e dei vari servizi OGC dispiegati
- Configurazione del motore di gestione dei dati raster
- Controllo dello stato di salute del GeoServer (visualizzazione log, controllo dell'utilizzo delle risorse)
- Gestione degli utenti e dei rispettivi diritti di accesso a servizi e risorse
- Preview dei dati configurati via client web

Dal punto di vista strettamente tecnico è opportuno sottolineare come la tecnologia usata per costruire la interfaccia grafica del GeoServer permetta di estendere la stessa in modo dinamico semplicemente dispiegando nuovi componenti che andranno ad integrarsi nella struttura di default. Questo permette di aggiungere nuove funzionalità in modo user-friendly senza stravolgere la interfaccia utente di base.

Supporto avanzato per la creazione di mappe⁷

Uno dei punti di forza del GeoServer è il supporto avanzato per il *rendering* di mappe tramite servizio WMS a partire da dati raster e vettoriali per mezzo dei file SLD (come da specifica OGC) in modo nativo ed estensivo, oltre ad una vasta gamma di estensioni specifiche.

Prima di proseguire è opportuno discutere brevemente lo scopo ed il contenuto dei file SLD. Un file SLD è un file XML che può essere usato per specificare direttive di *rendering* sia per dati vettoriali che per dati *raster*. Per quanto riguarda i dati vettoriali, tenendo conto del parallelo esistente fra una

⁷ Styled Layer Descriptor, è uno standard per specificare stili per GeoServer.

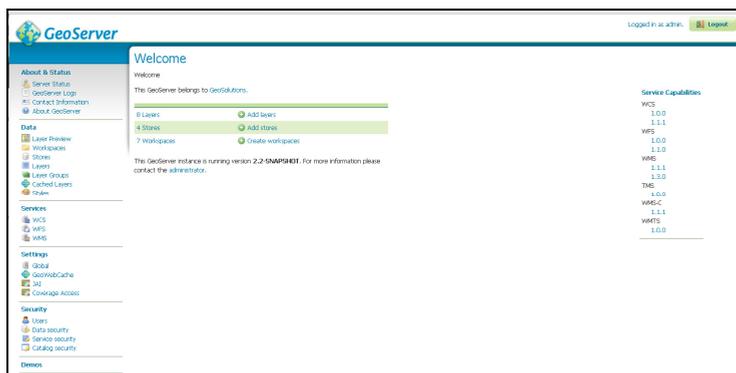


Figura 3 – La interfaccia grafica di amministrazione del GeoServer.

controllare il disegno delle geometrie in relazione ai valori degli attributi alfanumerici delle singole *feature* (si vedano le figure Figura 4 e Figura 5).

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<StyledLayerDescriptor version="1.0.0"
  xsi:schemaLocation="http://www.opengis.net/sld StyledLayerDescriptor.xsd"
  xmlns="http://www.opengis.net/sld"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xs="http://www.w3.org/2001/XMLSchema-instance">
  <NamedLayer>
    <Name>Simple Point</Name>
    <UserStyle>
      <Title>Simple Point</Title>
      <FeatureTypeStyle>
        <Rule>
          <PointSymbolizer>
            <Graphic>
              <Mark>
                <wellKnownName>circle</wellKnownName>
                <Fill>
                  <CssParameter name="fill">#FF0000</CssParameter>
                </Fill>
              </Mark>
              <Size>6</Size>
            </Graphic>
          </PointSymbolizer>
        </Rule>
      </FeatureTypeStyle>
    </UserStyle>
  </NamedLayer>
</StyledLayerDescriptor>
```

Figura 4 – Semplice SLD file per renderizzare un layer a punti.

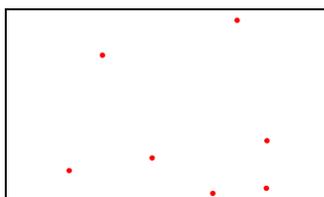


Figura 5 – Risultato del rendering controllato dal precedente SLD file.

Per quanto riguarda i dati *raster* i file SLD forniscono un supporto relativamente limitato per la loro manipolazione orientata al rendering, che comunque includono la possibilità di applicare mappe di colori personalizzati così come la possibilità di applicare funzioni di miglioramento del contrasto, come mostrato in Figura 6.

Nelle prossime sezioni andremo a descrivere brevemente alcune delle direttive di *rendering* più interessanti, siano esse supportate dagli standard WMS ed SLD oppure estensioni specifiche del GeoServer.

Supporto per Raster Symbolizer

GeoServer supporta la creazione di *rendering* artificiali a partire da raster che contengano valori geofisici (e.g. temperatura misurata sulla superficie del mare) tramite il meccanismo fornito dalla specifica SLD, introdotta sopra, detto *RasterSymbolizer*.

```
<RasterSymbolizer>
<ColorMap>
<ColorMapEntry color="#000000" quantity="0.0" opacity="1.0"/>
<ColorMapEntry color="#ff0000" quantity="2.0" opacity="1.0"/>
<ColorMapEntry color="#ffff00" quantity="10.0" opacity="1.0"/>
<ColorMapEntry color="#00ff00" quantity="15.0" opacity="1.0"/>
<ColorMapEntry color="#00ffff" quantity="20.0" opacity="1.0"/>
<ColorMapEntry color="#0000ff" quantity="25.0" opacity="1.0"/>
<ColorMapEntry color="#ff00ff" quantity="30.0" opacity="1.0"/>
<ColorMapEntry color="#ffffff" quantity="100.0" opacity="1.0"/>
</ColorMap>
</RasterSymbolizer>
```

Figura 6 – Esempio di applicazione di un *Raster Symbolizer* completo con selezioni banda, applicazione colormap, miglioramento del contrasto su singola banda e su immagine finale.

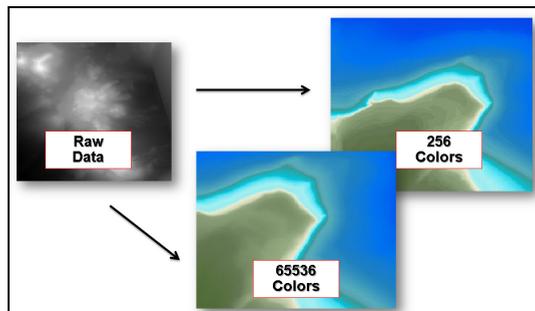


Figura 7 – *Raster Symbolizer* in azione per creare il rendering di un raster DEM.

Tramite questo meccanismo è possibile all'interno di un documento SLD istruire il servizio WMS del GeoServer a fare operazioni su di un raster quali selezione e merge di canali, *contrast stretch*, ma soprattutto applicazione di una *color map* specifica (a 256 o anche 65536 colori) tramite interpolazione lineare (di tipo *piecewise*) tra valori presenti nel raster e colori definiti.

Controllo avanzato delle etichette

GeoServer permette di controllare finemente, anche e soprattutto tramite varie estensioni allo standard SLD, il posizionamento delle etichette (*label*) per le varie geometrie supportate per i dati vettoriali anche e soprattutto in ragione degli attributi alfanumerici ad esse associate. E' opportuno sottolineare alcune fra le opzioni proprietarie a disposizione:

- *Priority Labeling*: permette di controllare l'ordine di preferenza tra etichette in relazione al valore degli attributi alfanumerici di una feature. Ad esempio (si veda Figura 8 e **Errore. L'origine riferimento non è stata trovata.**) è possibile chiedere al server di evidenziare le etichette delle città con un numero di abitanti superiore ad un certo valore.
- *Overlapping and Separating Labels*: permette di controllare finemente se e come le etichette possono sovrapporsi (si veda Figura 8 e **Errore. L'origine riferimento non è stata trovata.**).

Geometry Transformations

Il meccanismo delle *Geometry Transformations* ci permette di estendere il meccanismo di disegno delle geometrie per i dati vettoriali messo a disposizione dalla specifica SLD in modo da poter effettuare delle manipolazioni sulle geometrie stesse appena prima di applicare le direttive di

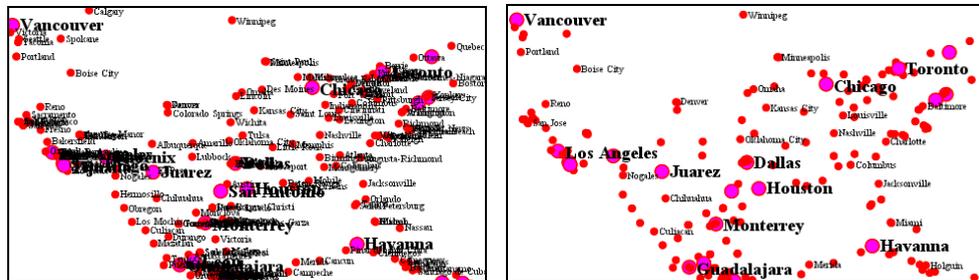


Figura 8 – Disegniamo tutte le città degli stati uniti, prima senza nessuna regola, poi controllando la priorità.

disegno: con questo meccanismo è possibile costruire manipolazioni complesse come quella mostrata in *Figura 9*.

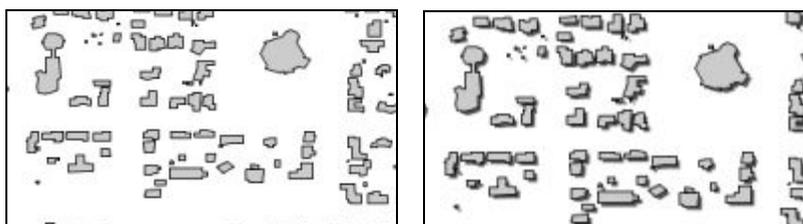


Figura 9 – Utilizzo delle *Geometry Transformations* per aggiungere una ombra a degli edifici.

Rendering Transformations

Il meccanismo delle *rendering transformations* può essere visto come una naturale estensione del meccanismo delle *geometry transformations* in quanto esso permette di applicare trasformazioni generiche su un intero tematismo prima che esso sia passato al motore di rendering e può essere applicato sia a dati vettoriali che a dati raster. Questo meccanismo si basa sui geoprocessi disponibili attraverso il servizio WPS, ossia possono essere usati come *rendering transformations* i geoprocessi installati dentro una singola installazione di GeoServer tramite installazione della estensione WPS.

Con questo meccanismo è possibile ad esempio creare uno stile che usa geoprocessi (sperimentali) di estrazione dei contorni e/o di poligonalizzazione di un raster ed allo stesso tempo indicare le direttive di disegno per le geometrie così generate, come mostrato in *Figura 10*.

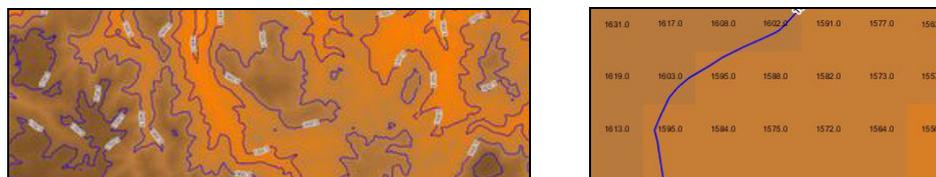


Figura 10 – Estrazione dei contorni da DEM, poligonalizzazione di DEM e loro disegno vettoriale on-the-fly.

Gestione della Qualità di servizio tramite Control Flow Module

Il modulo di control-flow è una estensione per GeoServer che fornisce all'amministratore gli strumenti per controllare il numero di richieste concorrenti che il server può gestire nonché la *Quality of Services (QoS)* ad esse riservata in termini di risorse utilizzate e tempi massimi di risposta. Questo tipo di controllo è importante per vari motivi:

- **Performance:** nostri test hanno evidenziato che limitare il numero di richieste WMS concorrenti a massimo 2 volte il numero di core CPU disponibile permette di incrementare visibilmente il *throughput* delle richieste di mappe.
- **Resource control:** richieste di estrazione dati o creazione mappe possono sfruttare grandi quantità di risorse di calcolo, e.g. memoria e CPU. La possibilità di imporre limiti sulle richieste, ad esempio sulle richieste di mappe, permette agli amministratori di controllare la quantità di memoria allocata ad ogni richiesta. Tramite i parametri di **Resource control** è però possibile controllare anche il numero totale di richieste servite contemporaneamente, controllando anche indirettamente la quantità massima di memoria utilizzabile dal GeoServer.
- **Fairness:** un singolo utente non dovrebbe sovraccaricare il server con troppe richieste, cannibalizzando le risorse del server stesso a discapito degli altri utenti.

Il modulo di control flow normalmente non rigetta le richieste in eccesso rispetto ai parametri di controllo, ma le accoda e le esegue ritardate nel tempo. E' comunque possibile configurare il modulo per rigettare le richieste dopo che esse siano state in coda per un tempo troppo lungo.

Supporto alla pubblicazione di KML⁸

GeoServer si integra agilmente con Google Earth supportando out-of-the-box innumerevoli feature interessanti riguardo alla produzione di KML, quali ad esempio:

- **SuperOverlays:** E' utilizzato nei casi in cui si devono rappresentare grandi quantità di dati e rappresenta una specifica forma di KML in cui i dati rappresentati sono suddivisi in regioni. Questo consente a Google Earth di richiedere o aggiornare ogni particolare regione della mappa solo quando questa viene visualizzata. Sono supportati i tipi di super-overlays raster e quello vettoriale. Il super-overlays raster produce delle immagini ottimizzate per il corrente livello di zoom della mappa e nuove immagini sono fornite quando questo cambia. Con il super-overlays i dati vettoriali vengono mostrati in modo progressivo, man mano che ci si avvicina alla superficie, per non sovraccaricare Google Earth.
- **Heights and Time templates:** Con GeoServer si ha la possibilità di definire dei modelli Freemarker (file height.ftl) per visualizzare i dati su Google Earth nel tempo e nella dimensione Z (di default Google Earth rappresenta i dati in due sole dimensioni). Configurando un modello di quota e scegliendo un attributo dal *layer*, si possono visualizzare su Google Earth le estrusioni delle singole feature sulla base dell'attributo scelto. Scegliendo invece di costruire un Time template (file time.ftl) si deve avere a disposizione, all'interno del dataset, un attributo temporale che, all'atto della visualizzazione di Google Earth viene utilizzato per la generazione di serie temporali sui dati.
- **Placemark templates:** modelli Freemarker possono essere utilizzati in GeoServer per personalizzare i Placemark di Google Earth. Si possono definire quindi due differenti file (title.ftl e description.ftl) il cui contenuto, sulla base sempre di uno o più attributi scelti, andrà a costituire il riempimento informativo del Placemark (si possono inserire anche elementi che indipendenti come immagini o link).
- **Regionation and scoring:** La modalità di super-overlay vettoriale può far uso della KML Regionation per organizzare le feature gerarchicamente. Questo richiede la specifica di un Regionation Attribute, che determina la modalità con cui le feature devono essere visibili ad

⁸ KML è il dialetto XML con cui le applicazioni geospaziali possono interagire con i client geospaziali di Google come Google Earth e Google Maps.

un particolare livello di zoom. GeoServer inoltre ha la possibilità di eseguire il rendering dei dati KML in formato raster o vettoriale. Questo viene determinato da quante feature sono presenti nella sorgente dati e dal valore dell'attributo KMSCORE.

Autenticazione

GeoServer fornisce di default supporto alla *HTTP Basic Authentication*⁹, sistema di autenticazione di larga diffusione e supportato da tutti i client OGC. Altri meccanismi di autenticazione possono essere integrati grazie alle funzionalità offerte da *Spring Security*, il modulo di sicurezza di Spring utilizzato da GeoServer anche se occorre prestare attenzione a quali client potranno beneficiare dell'uso di tali protocolli. Ad esempio, *Central Authentication Service (CAS)*¹⁰, un diffuso sistema di *Single Sign On*¹¹, permetterà una semplice integrazione con applicazioni web-based, ma impedirà l'accesso alla maggior parte dei client di tipo desktop.

Autorizzazione

GeoServer integra un sottosistema di sicurezza basato su *Spring Security* che permette agli utenti di impostare regole di autorizzazione sia a livello di servizio (e.g. regole che si applicano solo alle richieste WMS), sia regole di autorizzazione a livello di singolo *layer*.

Il sistema di sicurezza può essere suddiviso in due componenti:

- un motore di sicurezza di recente sviluppo che permette di applicare regole di sicurezza complesse filtrando sul singolo record, e nascondere singole colonne dati, nonché tagliare dati raster e *layer* WMS in cascata su specifiche aree di interesse
- una interfaccia di configurazione che permette di applicare regole più semplici a livello di singolo *layer* o singolo servizio.

Il motore permette di scrivere un sottosistema di sicurezza personalizzato e integrato con la gestione dell'autorizzazione del proprio dipartimento, mentre l'interfaccia di base permette di applicare semplici regole di sicurezza per installazioni non *enterprise*.

Accelerazione Disseminazione Mappe - Integrazione con GeoWebCache

GeoServer fornisce una integrazione trasparente con il framework geospaziale Open Source GeoWebCache per il caching dei *tile* e la conseguente accelerazione della disseminazione di mappe da server WMS verso client a-la Google Map. La funzionalità di GeoWebCache può essere equiparata a quella di un server proxy tra un client di mappe ed un server WMS. Si occupa sostanzialmente di gestire il caching dei *tile* così come questi sono richiesti salvando gli stessi su disco in una opportuna struttura di directory, eliminando in questo modo l'*overhead* introdotto da richieste ripetute, in quanto attraverso GeoWebCache i *tile* richiesti saranno salvati e recuperati dalla cache compatibilmente con i parametri specificati all'interno delle richieste WMS.

Supporto per il protocollo WPS

Il protocollo WPS, *Web Processing Service*, nasce per consentire la pubblicazione di operazioni di analisi spaziale in maniera interoperabile. Un servizio WPS può consumare dati vettoriali o raster, locali al server o remoti, e produrre una vasta gamma di operazioni comuni quali intersezioni, *buffering*, analisi statistiche su dati raster e vettoriali, algebra raster, conversioni di formato e molto altro. In effetti lo standard WPS impone le modalità di descrizione e chiamata dei processi senza però imporre la realizzazione di particolari funzioni di analisi spaziale, che sono invece lasciate alla discrezione di chi installa il server WPS. GeoServer di default fornisce un set di processi di partenza abbastanza esteso:

⁹ http://en.wikipedia.org/wiki/Basic_access_authentication

¹⁰ http://en.wikipedia.org/wiki/Central_Authentication_Service

¹¹ http://it.wikipedia.org/wiki/Single_sign-on

- decine di semplici processi di analisi su singole geometrie vettoriali insieme ad un discreto numero di processi su tematismi vettoriali quali *clipping*, *buffering* controllato da attributo, *overlay* vettoriale, unione e dissolve
- analisi statistiche su dati raster, raster contro raster classificato, o raster contro poligonale, clip di un raster su poligono, georeferenziazione di un raster dati i punti di controllo al suolo
- vari processi di conversione fra raster e vettoriale, inclusa la estrazione di isolinee

E' inoltre possibile scrivere nuovi processi con una API Java di facile utilizzo. Il supporto alla scrittura di processi in linguaggi di *scripting* quali Python, JavaScript, Scala o JRuby e Groovy è inoltre in fase di sviluppo. Da segnalare la profonda integrazione del servizio WPS con il resto delle funzionalità fornite dal server, citata in precedenza a proposito delle *Rendering Transformations*.

GeoServer Facts and Numbers

GeoServer è supportato da una comunità di sviluppo e supporto vibrante ed attiva. Inoltre supporto commerciale è disponibile attraverso le seguenti compagnie:

- [GeoSolutions](#) (Italia) è un società operante nel settore della Information & Communication Technology (ICT) geospaziale con la missione di fornire servizi di supporto professionale e soluzioni customizzate innovative, robuste e cost-effective basate sulle migliori tecnologie Open Source. Gli associati di GeoSolutions sono membri del PMC¹² di GeoTools, del PSC¹³ di GeoServer ed anche del PSC di GeoNetwork (Geo20).
- [OpenGeo](#), la divisione geospaziale di [The Open Planning Project \(TOPP\)](#) (US), una organizzazione no-profit con sede a New York.

Project Steering Committee

Il progetto Open Source GeoServer è governato dal Project Steering Committee (PSC). Tale organismo è formato da persone individuali con lo scopo di rappresentare quanto più possibile le diverse componenti della comunità di sviluppatori e power users del GeoServer stesso. Il turnover è ammesso ed atteso per garantire un generale ricambio delle persone che siedono nel PSC in modo lasciare spazio mano mano alle nove risorse che emergono dalla comunità. Il ruolo primario di un membro del PSC è di prendere decisioni riguardo alla gestione del progetto GeoServer, sia operativa che strategica.

Diffusione

Essendo un progetto Open Source liberamente scaricabile dal quindi una buona sorgente di informazioni per misurare la popolarità del GeoServer è rappresentata dalle statistiche dei download dal sito di distribuzione (SourceForge). Nel periodo Settembre 2009-Settembre 2010 il numero medio di download al mese è stato di 20000. Inoltre GeoServer è attualmente dispiegato in molti ambienti differenti che vanno da aziende private commerciali fino a grandi istituzioni sovranazionali quali FAO (vari dipartimenti), IFAD, WHO, World Bank, Protezione Civile Italiana, Regione Sardegna, Regione Friuli, Regione Calabria.

Bibliografia

- Bedward, Michael. 2009.** JAITools, Raster image processing for Java developers. *JAITools, Raster image processing for Java developers*. [Online] 2009. [Cited: Agosto 3, 2011.] <http://code.google.com/p/jaitools/>.
- Butler , Howard , et al. 2008.** The GeoJSON Format Specification. *The GeoJSON Format Specification*. [Online] June 16, 2008. <http://geojson.org/geojson-spec.html>.

¹² Project Management Committee

¹³ Project Steering Committee

- Davis, Martin. 2002.** *JTS Topology Suite. JTS Topology Suite.* [Online] Refrations, 2002. [Cited: Agosto 30, 2011.] <http://sourceforge.net/projects/jts-topo-suite/>.
- de La Beaujardière, Jeff and Doyle, Allan. 2002.** *Web Map Service Implementation Specification.* s.l. : Open GIS Consortium Inc., 35 Main Street, Suite 5, Wayland, MA, USA, 2002.
- de la Beaujardiere, Jeff. 2006.** *OpenGIS® Web Map Server Implementation Specification, Version 1.3.0.* Wayland, MA, USA : Open Geospatial Consortium Inc., 2006. OGC® 06-042.
- ESRI. 1998.** ESRI Shapefile Technical Description. *ESRI Shapefile Technical Description.* [Online] Luglio 1998. [Cited: Agosto 30, 2011.] <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>.
- Evans, John D. 2003.** *Web Coverage Service (WCS), Version 1.0.0.* Wayland, MA, USA : Open Geospatial Consortium Inc., 2003. OGC 03-065r6.
- Free Software Foundation. 1991.** GNU GENERAL PUBLIC LICENSE, Version 2.0. *GNU GENERAL PUBLIC LICENSE, Version 2.0.* [Online] 1991. [Cited: Agosto 30, 2011.] <http://www.gnu.org/licenses/gpl-2.0.txt>.
- GeoNetwork opensource. *GeoNetwork opensource.* [Online] [Cited: Agosto 2011, 20.] <http://geonetwork-opensource.org/>.
- GeoRSS Website. *georss.org.* [Online] http://georss.org/Main_Page.
- GeoSolutions. 2007.** Imageio-ext. *Imageio-ext.* [Online] GeoSolutions, 2007. [Cited: Agosto 30, 2011.] <http://java.net/projects/imageio-ext>.
- GeoTools The Open Source Java GIS Toolkit. *GeoTools The Open Source Java GIS Toolkit.* [Online] [Cited: Agosto 30, 2011.] <http://geotools.org/>.
- Lalonde, William. 2002.** *Styled Layer Descriptor Implementation Specification.* s.l. : Open GIS Consortium Inc., 2002. OGC 02-070.
- Muller, Dr. Markus. 2006.** *Symbology Encoding Implementation Specification, Version 1.1.0.* Wayland, MA, USA : Open Geospatial Consortium Inc., 2006. OGC 05-077r4.
- Oracle. 2001.** Jai-core. *Jai-core.* [Online] Oracle, 2001. [Cited: Agosto 30, 2011.] <http://java.net/projects/jai-core>.
- . 2011. Jai-imageio-core. *Jai-imageio-core.* [Online] Agosto 30, 2011. <http://java.net/projects/jai-imageio-core>.
- Schut, Peter. 2007.** *Web Processing Service, Version 1.0.0.* Wayland, MA, USA : Open Geospatial Consortium Inc., 2007. OGC 05-007r7.
- Vretanos, Panagiotis. 2005.** *Filter Encoding Implementation Specification, Version 1.1.0.* Mayland, MA, USA : Open Geospatial Consortium Inc., 2005. OGC 04-095.
- . 2002. *Web Feature Service Implementation Specification.* Wayland, MA, USA : Open GIS Consortium Inc., 2002. OGC 02-058.
- . 2005. *Web Feature Service Implementation Specification, Version 1.1.0.* Wayland, MA, USA : Open Geospatial Consortium Inc., 2005. OGC 04-094.
- Whiteside, Arliss and Evans, John. 2006.** *Web Coverage Service (WCS) Implementation, Version 1.1.0.* Wayland, MA, USA : Open Geospatial Consortium Inc., 2006. 06-083r8.
- Wilson, Tim. 2008.** *OGC® KML.* Wayland, MA, USA : Open Geospatial Consortium Inc., 2008. OGC 07-147r2.