

Dalle banche dati geografiche al Web: un approccio avanzato allo scambio ed alla condivisione delle informazioni

Marco CIOLLI, Alfonso VITTI, Paolo ZATELLI, Fabio ZOTTELE

Dipartimento di Ingegneria Civile e Ambientale
Università di Trento
Via Mesiano 77, 38100 Trento, Italy
tel. +39 0461 882618, fax +39 0461 882672
e-mail paolo.zatelli@ing.unitn.it

Introduzione

Il lavoro qui presentato descrive la pubblicazione via web di dati di flussi di traffico veicolare, inquinamento e qualità dell'aria nel Comune di Trento.

Per la condivisione di tali informazioni è stato approntato un sistema di tecnologie software Open Source che garantisca all'utente un servizio trasparente di consultazione dei dati e di semplice utilizzazione. Qualora l'utente decidesse di elaborare i dati, sia geografici che semantici, questi vengono a lui forniti in formati standard pronti per il download ed in formato compatibile con i più diffusi software di visualizzazione e di analisi GIS.

La scelta di rivolgersi ad una soluzione di tipo Open Source permette di ottenere una riduzione dei costi ed offre soluzioni software tecnologicamente avanzate, continuamente aggiornate, mature, modulari, scalabili e che soddisfano pienamente le esigenze già evidenziate. L'architettura del sistema è basata sui programmi per la gestione delle banche quali PostgreSQL, accoppiato con l'estensione spaziale PostGIS e sulla capacità che MapSERVER ha, attraverso la libreria php_mapscript, di produrre mappe web processando direttamente i dati grezzi contenuti nel database. La migrazione dei dati geometrici e semantici verso i computer dei client per eventuali elaborazioni successive è stata risolta producendo output in formato GML (Geometry Markup Language) come descritto dalle specifiche dell'OpenGeospatial Consortium, appoggiandosi alla libreria OGR. Da ciò scaturisce la massima condivisione delle informazioni. Un ruolo ancillare rivestono numerose librerie di trattamento dei dati e di gestione della topologia, mentre per lo sviluppo delle applicazioni necessarie ci si è avvalso di PHP5, affiancato al Webserver Apache, e di javascript.

L'architettura software è stata armonizzata per ottenere una sinergia tra i vari elementi massimizzandone non solo l'efficacia, ma anche l'efficienza.

Abstract

The work describes the web publication of traffic fluxes, air pollutant concentration and air quality for the city of Trento.

For sharing such data, various Open Source softwares have been set up to provide a transparent, clear and efficient service to the end user. When the user needs to download data, the chosen thematic layers are converted on the fly in GML standard format and made available to the user.

The Open Source approach reduces the amount of monetary resources while offering a technologically advanced, cutting edge solution, thus granting scalability and reliability. System architecture is based on a splinter approach with PostgreSQL in synergy with Mapserver. The conversion to the standard GML format is managed by OGR. This fixes the compatibility issue granting the state of art in information division. The software architecture based on PHP, Javascript, Apache and ancillar libraries for geographic data management has been tuned for

maximizing not only effectiveness but performance too.

Identificazione del problema

Rendere disponibili al pubblico dati georeferenziati è una sfida che, negli ultimi anni, ha dato vita al fiorire di soluzioni tecnologiche avanzate quali, ad esempio, Google Earth, Google Map, Yahoo! Maps... Gli approcci al problema partono dalla domanda di base che nel mondo dei servizi di web mapping suona in questo modo: *client side* o *server side* ?

Tradizionalmente i servizi di WebGIS erano, e talvolta rimangono, di tipo *client side*: attraverso l'utilizzo di plugin da installare sulla macchina dell'utente è possibile effettuare una connessione ad un database centralizzato, scaricare sul computer client tutte le informazioni necessarie ed in seguito visualizzarle sul browser. Il vantaggio di questo approccio risiede nella velocità: i dati vengono spediti grezzi ed elaborati in locale dalla macchina del client: la navigazione consiste nel modificare e riarrangiare le informazioni al fine di visualizzarle a schermo.

L'approccio *server side* consiste nell'interpretare le richieste che il *client* genera ad un server remoto: il WebServer crea una mappa da inviare fino al browser del client. Ogni volta che il client effettua un'operazione di zoom o l'aggiunta di un tematismo, una nuova mappa viene elaborata dal server ed inviata. L'approccio è più macchinoso, ma ha un indiscutibile vantaggio: il client può utilizzare il browser senza dover scaricare plugin, talvolta a pagamento, per la visualizzazione, o preoccuparsi di aggiornare il software di visualizzazione.

Nell'applicazione qui presentata si è preferito l'approccio *server side* in maniera da rendere il servizio perfettamente trasparente al client: nessun programma aggiuntivo deve essere installato sul computer di chi richiede le mappe, a meno di un browser web. Questo approccio svincola il client ma richiede particolare attenzione nell'allestimento dell'architettura software *server side*.

Innanzitutto occorre pensare dove immagazzinare le informazioni geografiche ed in quale formato: la risposta dipende da quali formati siano supportati dal software che produrrà la mappa da inviare al client. Nel mondo Open Source è, attualmente, fortissima la spinta alla standardizzazione dei tipi di dato geometrico e della loro rappresentazione (Open Geospatial Consortium, 1999). Ne consegue che i programmi (commerciali e non) che supportano tale tipo di standard riescono a trasferire le informazioni in maniera efficiente armonizzando il processo di trasferimento delle informazioni.

I dati di flussi di traffico e la metodologia per il calcolo delle concentrazioni di inquinanti nell'atmosfera risiedono in un database PostgreSQL (The PostgreSQL Software Development Group, 2006). Risulta quindi naturale applicare al database già esistente l'estensione spaziale PostGIS (Refraction Research, 2006): esso fornisce funzionalità e tipi geometrici per la gestione dei dati spazializzati ed è in accordo con le specifiche OGC. PostgreSQL è inoltre conforme al protocollo ACID (Atomicity, Consistency, Isolation, Durability) che garantisce la sicurezza della coerenza dei dati e che fornisce garanzie sul meccanismo dell'uso concorrente dei dati da parte di più utenti. In questa maniera più client possono richiedere i dati da mappare ed i software possono agire su di essi contemporaneamente gestendo l'ordine di accesso concorrente. Inoltre, la robustezza del database garantita da PostgreSQL fornisce l'ambiente elettivo per testare l'interoperabilità dei GIS (commerciali e non) con repository di dati centralizzati (Vallée R., 2004). Su questo tema gli autori stanno conducendo test per valutare l'effettiva rispondenza l'interoperabilità dei maggiori GIS commerciali verso PostgreSQL/PostGIS.

Un database robusto come PostgreSQL può inoltre supportare massicci scambi di dati e pesanti carichi computazionali senza perdere efficienza nella gestione. Per questo motivo è quindi pensabile di utilizzare tale software per immagazzinare dati raccolti in tempo reale.

Occorre, a questo punto, identificare quale tecnologia software risulti migliore per garantire il servizio di Web Mapping partendo dai dati presenti nell'archivio spaziale e restituendo cartografia interattiva.

Open Source

Sebbene il database centrale dei dati semantici e geometrici sia un prodotto Free e Open Source, nulla vieterebbe di costruire un servizio di WebMapping utilizzando software Free, ma con codice non aperto, o soluzioni totalmente commerciali. I motivi che hanno indotto gli autori a scegliere software Open Source sono molteplici. Innanzitutto, nell'ambiente della ricerca è fondamentale disporre del codice sorgente del software: in questa maniera è possibile conoscere a fondo le potenzialità dei mezzi a disposizione e sviluppare applicazioni partendo da essi. Inoltre la comunità degli sviluppatori di software Open Source è disponibile a condividere le loro esperienze, al confronto per migliorare i prodotti e a sostenere gli utenti in difficoltà. Inoltre i progetti Open Source risultano essere maturi e vengono utilizzati massicciamente nell'ambito della produzione per la loro riconosciuta stabilità ed efficienza. La possibilità di avere tanti software specializzati per effettuare un compito ben preciso permette di modulare un'architettura complessa giovandosi della sinergia tra i vari elementi. Quale valore aggiunto di questa scelta si ha un risparmio economico dal punto di vista dell'acquisto di licenze potendo investire maggiori fondi in hardware più potente e in tecnici per lo sviluppo degli applicativi.



Illustrazione 1: webGIS per la mappatura ed il download di flussi di traffico e concentrazione di inquinanti atmosferici. Attraverso gli strumenti di navigazione è possibile il download dei tematismi visualizzati

Il servizio di web mapping

Avendo scelto di utilizzare un approccio *server side* occorreranno una serie di metodologie che interpretino le richieste che il client invoca attraverso il web server e che trasformino dette richieste in una mappa da inviare al client. La produzione della mappa avviene utilizzando Mapserver e, specificatamente la libreria *php_mapscript* (McKenna J., 2006) che si integra nel linguaggio di scripting PHP per la creazione di pagine HTML. Mapserver si interfaccia perfettamente con PostGIS attraverso l'applicazione degli standard OGC (McKenna T., Mitchell T., 2005). Negli ultimi anni, inoltre, si sono sviluppati una serie di prodotti che, basandosi su Mapserver, consentono di semplificare e rendere più spedita la produzione della pagina che il client visualizzerà sul browser. Nel caso qui riportato si è scelto Chameleon (Redfern D., 2006): una collezione di script PHP e Javascript che consentono di inserire gli elementi di navigazione, interrogazione e la mappa

stessa direttamente nel codice HTML. Ciò ha consentito di sviluppare in maniera più rapida l'interfaccia client e di potersi concentrare sulla funzionalità di download dei dati visualizzati in formato GML (Open Geospatial Consortium, 2006) per la successiva elaborazione dei dati scaricati con programmi GIS installati localmente sul computer client. Per integrare questa funzionalità nel webGIS ci si è appoggiati su OGR Simple Feature Library (Warmerdam F., 2005) che permette di trasformare le informazioni immagazzinate nel database di PostgreSQL/PostGIS in file di formato GML. Il client, scegliendo i tematismi, zoomando e navigando nella mappa interattiva, definisce una regione di interesse: le entità geometriche che ricadono all'interno di quella regione verranno trasformate in features GML e i dati semantici collegati ad esse (tematizzazioni) verranno inglobate nel file stesso attraverso OGR. In questa maniera il client effettuerà il download dei soli dati di suo interesse invece di immagazzinare sul computer una grande quantità di informazioni inutili che rallenterebbero tutta la procedura.

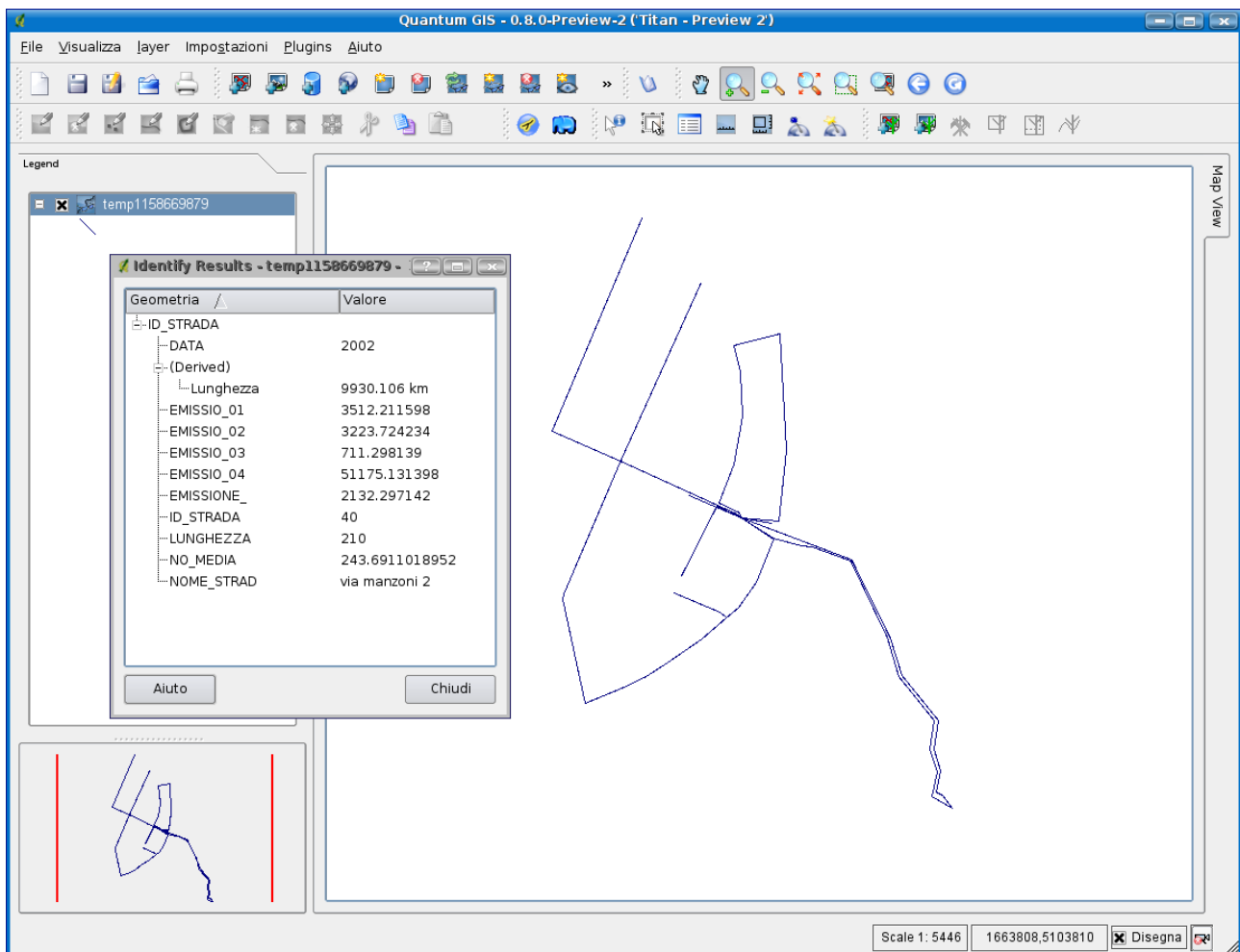


Illustrazione 2: Restituzione sul GIS del client dei dati in formato GML downloadati.

Architettura software

<i>software</i>	<i>versione</i>	<i>utilizzo</i>
Ubuntu Server	6.06	sistema operativo
PostgreSQL	8.1.4	dati semantici: flussi di traffico, modello CopertIII, qualità dell'aria
PostGIS	1.1.3	estensione spaziale a PostgreSQL

<i>software</i>	<i>versione</i>	<i>utilizzo</i>
Apache	2.0.55	web server
PHP	5.1.2	server side scripting language
Mapserver	4.8.4	WebGIS
Chameleon	2.4.1	ambiente di sviluppo per applicativi web mapping
OGR	1.3.2	Simple Feature Library

Tabella 1: Software e versioni utilizzate

Conclusioni

Il lavoro qui presentato riesce a soddisfare gli obiettivi prefissati per la realizzazione del progetto. Occorre qui sottolineare che il mondo dei *Web Mapping Services* è in continuo fermento introducendo rapidamente innovazioni tecnologiche per rendere le interfacce web sempre più semplici, veloci e navigabili al fine di dare al client soluzioni molto appetibili. Nel campo della ricerca occorre quindi intraprendere studi approfonditi sul *cross browsing* ovvero testare il servizio su tutti i tipi di browser (o almeno i più diffusi). Accade di frequente che alcuni Javascript, forniti assieme agli ambienti di sviluppo, non vengano supportati su alcune piattaforme di navigazione (Opera, Safari, Konqueror) e che quindi venga negato l'utilizzo del servizio ad una fascia, benchè ristretta, di utilizzatori.

Come in ogni percorso di ricerca, durante l'opera si sono rivelate nuove possibilità e strategie per migliorare il servizio. Tra queste l'interazione diretta tra GIS (tecnologie proprietarie e non) e Database Spaziale centralizzato e l'aggiunta di nuovi formati di interscambio di dati (ESRI shapefile, Mapinfo file, Autodesk dwg file) disponibili per il download. Un ulteriore ambito di ricerca rappresenta lo scambio di dati direttamente tra database spaziali (MySpatial, OracleSpatial, ArcSDE...).

Inoltre è possibile ripensare i servizi da proporre all'utente affiancando al PHP la tecnologia AJAX (Asynchronous Javascript + XML): essa è massicciamente utilizzata da servizi di web mapping proprietari quali Google Map o Yahoo!Map. La grande diffusione di queste tipologie di servizio ha reso familiari all'utenza i servizi di web mapping per la loro semplicità e velocità. Il valore aggiunto della sinergia tra AJAX e PHP risiede nella maggiore inerattività delle pagine web che non deve, però, andare a scapito della effettiva efficienza del servizio.

Sebbene gli ambienti di sviluppo di applicazioni di web mapping possano risultare di aiuto per la produzione di webGIS semplici, la loro struttura, a volte troppo rigida, non permette in alcuni casi di affiancare alle funzionalità di base applicazioni più evolute (il download dei dati ne è un esempio). Si potrebbe ipotizzare di cambiare l'approccio al problema e partire dalla ridefinizione della pagina web non utilizzando ambienti di sviluppo precostruiti, ma integrando PHP/mapscript con AJAX direttamente. Ciò implica d'altra parte che parte delle risorse dovranno essere impiegate nella programmazione delle API necessarie. La ricerca di un accordo sulla standardizzazione degli ambienti di sviluppo è tuttavia una richiesta pressante anche da parte degli stessi sviluppatori.

Ringraziamenti

Questo lavoro è stato svolto grazie ai fondi del Ministero Italiano dell'Università e della Ricerca (MIUR) nell'ambito del progetto MIUR-COFIN2004: strutture evolute della cartografia numerica per i GIS e l'ambiente Web.

Un ringraziamento particolare va a Julien Samuel Lacroix (MapGear), Tayler Mitchell (*MI05*) ed a tutta la comunità degli sviluppatori di PostGIS, MapServer, PHP e Chameleon.

Riferimenti bibliografici

Darren R. (2005), "Chameleon Application Developer's Guide - version 2.0",
<http://chameleon.maptools.org/help/>

Open Geospatial Consortium (1999), "OpenGIS Geometry MarkUp Language (GML) Encoding Specifications", <http://www.opengeospatial.org/standards/gml>

McKenna J., Mitchell T. (2005), "MapServer Reference Guide for Vector Data Access",
http://mapserver.gis.umn.edu/docs/reference/vector_data

McKenna J. (2006), "PHP/Mapscript Class reference",
<http://mapserver.gis.umn.edu/docs/reference/phpmapscript-class>

Mitchell T. (2005), "Web Mapping Illustrated", *o'Reilly*, ISBN: 0-596-00865-1

Open Geospatial Consortium (1999), "OpenGIS Simple features Specification for SQL",
<http://www.opengis.org/docs/99-049.pdf>

Warmerdam F. (2005), "OGR - Feature Style Specification",
http://ogr.maptools.org/ogr_feature_style.html

Refraction Research Inc. (2006), "PostGIS Manual", <http://postgis.refractory.net/docs/postgis.pdf>

Achour M., Betz F., et al. (2006), "PHP Manual", <http://www.php.net/manual/en/preface.php>

The PostgreSQL Development Group (2006), "PostgreSQL 8.1.4 Documentation",
<http://www.postgresql.org/docs/8.1/interactive/index.html>

Vallee R. (2004), "SIG et SGDB: Etude sur l'interconnexion et l'interopérabilité de Oracle 9i et de PostgreSQL",
http://www.carto.net/papers/romain_valee/romain_valee_interoperabilite_oracle9i_PostgreSQL?070704