

BRISEIDE - implementazione di servizi OGC con dimensione temporale

Gianni Barrotta (*), Piergiorgio Cipriano (**), Stefano Pezzi (***)

(* Sinergis srl, Piazza della Repubblica 32, Catania (CT), Tel. +39.095.312982, gianni.barotta@sinergis.it

(**) Sinergis srl, Via del Lavoro 71, Casalecchio di Reno (BO), Tel. +39.051.278911, piergiorgio.cipriano@sinergis.it

(***) Sinergis srl, Via del Lavoro 71, Casalecchio di Reno (BO), Tel. +39.051.278911, stefano.pezzi@sinergis.it

1. Introduzione

La gestione di rischi ed emergenze ambientali è alla base del progetto europeo BRISEIDE (<http://www.briseide.eu/>).

L'enfasi del progetto è posta sulla componente "temporale" dei dati geografici (sia vettoriali che raster); di conseguenza, per l'utilizzo efficiente di tali dati con servizi OGC è necessario avere a disposizione strumenti e tecnologie in grado di:

- caricare automaticamente dati dinamici (es. rilievi meteorologici e previsioni) e statici (es. uso del suolo)
- catalogare automaticamente su server OGC sia dati che relativi metadati
- pubblicare servizi OWS con funzioni "spazio-temporali"

I casi d'uso raccolti dai 10 pilot del progetto hanno infatti evidenziato la necessità di avere capacità elaborative con filtri temporali per:

- il *portrayal* di dati vettoriali e *coverage* (WMS)
- l'accesso ed il *download* di dati vettoriali (WFS) e *coverage* (WCS)
- il *processing* e la trasformazione di dati (WPS)

La soluzione architetturale del progetto è basata completamente su soluzioni *open source*:

- GeoBatch per l'*ingestion* e la catalogazione automatica di dati su GeoServer e dei relativi metadati su GeoNetwork
- GeoServer per il *portrayal* (WMS), l'accesso (WFS, WCS) ed il processamento di dati (WPS)
- PostGres + PostGIS per il database (dati vettoriali e metadati)
- GeoTools
- GDAL

2. Modello dati e dimensione temporale

Briseide ha affrontato il problema della dimensione temporale nei dati geografici nei due ambiti tipici in cui si classificano abitualmente i dati geografici tradizionali: raster e vettoriale. In entrambi i casi sono stati definiti dei modelli dati per supportare il trattamento dell'attributo temporale da parte dei servizi di accesso e di mappa. Nel caso dei dati vettoriali una condizione vincolante è stata quella del formato di memorizzazione che il progetto ha individuato a priori nel DB relazionale spaziale PostGIS, mentre per i dati raster non sono state fatte scelte particolari, visto anche il contesto più variegato dei formati esistenti per questo tipo di dati. In tutti e due i casi però si è limitato l'ambito di intervento del progetto, escludendo nuovi sviluppi che riguardassero direttamente il DBMS o la creazione di nuovi formati raster

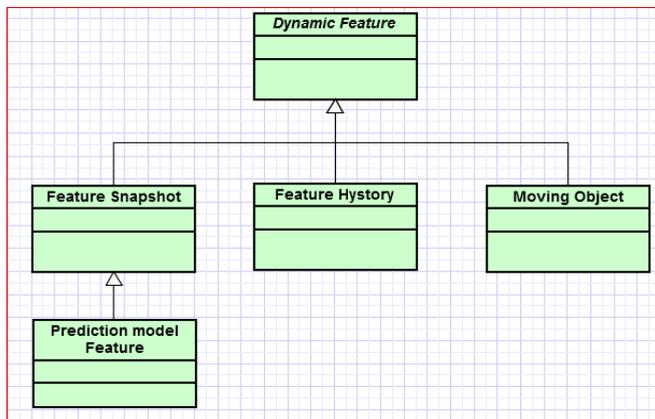
In entrambi i casi si sfrutterà il disegno architeturale del sistema che prevede la presenza di servizi di ingestione, che possono implementare la logica necessaria a trasformare i dati di input nei modelli implementativi definiti.

Dati vettoriali

In Briseide sono stati individuati tre tipologie di dati spazio-temporali modellabili come dati vettoriali:

- *Feature History* (oggetti con storia)
- *Feature Snapshot* (oggetti istantanei)
- *Moving Object* (oggetti in movimento)

Tutti questi oggetti modificano le loro proprietà nel tempo, inclusa la geometria, e potrebbero in teoria essere ricondotti ad un unico modello di oggetto spazio-temporale o *Dynamic Feature*. Nel contesto del progetto si è preferito ricorrere a questa suddivisione che corrisponde ad una pratica categorizzazione dei casi d'uso individuati e che risponde più appropriatamente alle esigenze dei diversi utenti. Nella figura sottostante è indicata la gerarchia delle classi di oggetti spazio temporali individuate (si può notare anche una specializzazione della *Feature Snapshot* che sarà introdotta più avanti).



Tipologie di attributi temporali

La tipologia di dimensione temporale considerata fino ad ora ha sempre avuto un solo significato: il tempo in cui una proprietà di una *Dynamic Feature* ha cambiato valore nel mondo reale. Per essere corretti invece occorrerebbe considerare diversi possibili significati dell'attributo temporale; infatti possiamo parlarne di:

- *valid time* (tempo di validità o tempo intrinseco o tempo del mondo reale o tempo logico). Il valid time di un fatto è il tempo in cui il fatto è vero nella realtà che si modella.
- *transaction time* (tempo della transazione o tempo estrinseco o tempo di registrazione o tempo fisico). E' il tempo in cui un "fatto del database" (che descrive un fatto del mondo reale) è memorizzato nel database. Non si può inserire un transaction time futuro né si può modificare un transaction time passato.
- *user-defined time* (tempo utente). E' un attributo che ha sì il dominio di valori di un temp (data e ora), ma che non assume nessun significato particolare nell'analisi temporale e nella gestione del dato; in altre parole un user-defined time è allo stesso livello di un qualsiasi altro attributo con dominio di valori differente (ad esempio, la data di collaudo di un componente meccanico).

Queste tipologie di attributi temporali sono applicabili sempre e a tutti i tipi di oggetto spazio-temporale; per particolari tipologie di oggetti si possono aggiungere attributi temporali con specifici significati, come, ad esempio, il *run time* (tempo di esecuzione del modello predittivo) per le feature prodotte da modelli matematici di previsione.

Feature History

Sono oggetti che cambiano la loro posizione o geometria e un qualsiasi altro attributo nel tempo. Possono essere:

- oggetti fisici, naturali o artificiali, semplici (un edificio, un fiume ...) o complessi (un insediamento, una foresta ...).
- fenomeni naturali o artificiali (un incendio, una tempesta, uno sversamento di petrolio in mare,...), cioè entità rappresentate da un'area che non è limitata da una confine ben definito.

Alcuni oggetto cambiano continuamente nel tempo, altri cambiano solo in coincidenza di particolari eventi che avvengono in istanti specifici. Per i primi normalmente si ha un tempo di campionamento (più o meno costante) ed i valori relativi campionati (immagini satellitari di un uragano, la serie temporale delle foto aeree di un parco naturale...); per gli ultimi si hanno informazioni solo relative a questi particolari eventi (la data dei documenti che autorizzano il cambiamento di sagoma di un edificio, un rilievo di un fiume dopo un episodio di straripamento).

E' importante sottolineare che un singolo oggetto mantiene la sua identità individuale attraverso i vari cambiamenti di stato per ragioni intrinseche o convenzionali: se un edificio aumenta la sua superficie o un fiume modifica il suo corso è banale capire che l'edificio è ancora lo stesso oggetto e che il fiume è ancora lo stesso fiume. Questo succede perché la definizione di cosa è chiamato fiume è molto chiara e per l'edificio sono ben definite le regole che determinano il suo ciclo di vita. Ad esempio, potremmo decidere che quando un edificio viene spezzato in due o più fabbricati, non si può più considerare lo stesso oggetto; stessa cosa se un edificio viene demolito e ricostruito anche se con la stessa forma.

L'interesse principale per l'utente è nella geometria e nella posizione dell'oggetto e anche in tutti gli altri attributi alfanumerici.

Feature Snapshot

Simili alle *Feature History*, ma con la differenza che tra uno stato e l'altro non è possibile dire quale oggetto si è trasformato in quale. Non è riconoscibile nemmeno un rapporto di parentela tra le feature e questo perché nessuna feature possiede in realtà una vera e propria identità anche se non consideriamo la dimensione temporale. Un esempio efficace può essere quello di mappe prodotte dalla classificazione automatica di immagini raster: qui i poligoni non si riferiscono ad un particolare oggetto fisico, ma a *pixel* adiacenti che hanno i valori delle diverse bande di frequenza che ricadono nello stesso intervallo. Quando classifichiamo la stessa area in tempi differenti, i nuovi poligoni non sono correlati a quelli vecchi solo per essere nella stessa classe o condividere la medesima porzione di territorio. Quindi è come se tutti gli oggetti precedenti venissero cessati e i nuovi venissero creati.

Prediction Model Feature

Per alcuni tipologie di *feature* spazio-temporali è opportuno considerare un'altra dimensione temporale. E' il caso delle simulazioni o delle previsioni derivanti dall'esecuzione di un modello matematico predittivo. La dimensione è il *run time*, ovvero il tempo di esecuzione del modello. Le feature prodotte da modelli predittivi sono solitamente *Feature Snapshot* dove ciascun singolo oggetto prodotto da un'esecuzione non ha nessuna relazione con le *feature* prodotte dall'esecuzione precedente o successiva.

Moving object

Sono oggetti che cambiano la loro posizione e altri attributi nel tempo. La loro geometria (forma) è sempre fissa e, per i nostri scopi, può essere rappresentata da un semplice punto. E' un oggetto fisico, spesso artificiale, che cambia posizione nel tempo perché ha capacità autonome di movimento (un veicolo, un animale, una barca...) o perché non è fissato al terreno ed è soggetto al moto del mezzo nel quale è immerso (una boa galleggiante, un pallone meteorologico,...). Normalmente questi oggetti cambiano continuamente la loro posizione e i dati sono basati su un periodo teoricamente costante di campionamento (un trasmettitore che spedisce i dati ogni 5min, un video o una sequenza di immagini che riprendono l'oggetto in movimento). L'interesse principale dell'utente è per la posizione stessa dell'oggetto e opzionalmente in qualche parametro relativo al moto (velocità, accelerazione). Altri attributi relativi all'oggetto o all'ambiente circostante possono essere di interesse (livello della benzina nel serbatoio, la frequenza cardiaca dell'animale monitorato, la temperatura dell'acqua...).

3. Dati raster

Un raster è una struttura complessa che descrive un insieme di *dataset* matriciali che hanno lo stesso dominio e stessa configurazione di bande. Solitamente può essere il risultato di osservazioni con sensori remoti, fotografie o esecuzioni di modelli matematici. Un raster multidimensionale, dove alle tipiche dimensioni X, Y della griglia si aggiungono, ad esempio, quella temporale (o quelle temporali, se ad esempio consideriamo anche il tempo di esecuzione di un modello, o run time) e l'elevazione, può essere schematizzato dalla figura seguente:

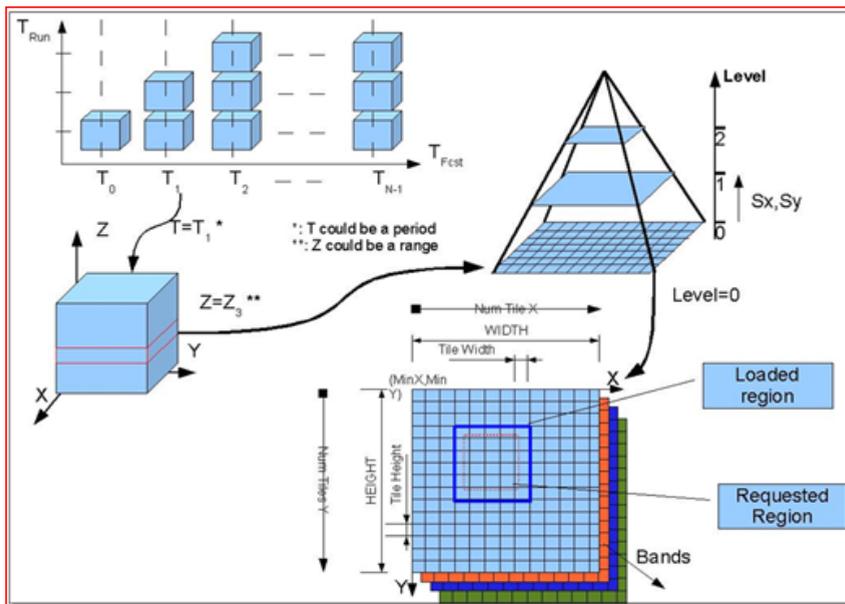


Figura 1 – Modello dati raster.

dove il primo livello di aggregazione è il tempo. Nel caso più generale l'ipercubo di dati può avere delle sovrapposizioni nel tempo, come ad esempio mostrato nel grafico in alto a sinistra dell'immagine (i raster sono in questo caso il prodotto di un modello di previsione che viene fatto girare in successivi T_{run} , producendo degli ipercubi di dati che si riferiscono a diversi T_{fcst} ; per lo stesso istante quindi posso avere più previsioni generate in tempi di esecuzioni diversi). Nel nostro

caso ipotizziamo che i dataset sia un insieme di fette (*slice*) costituite da raster bidimensionali corrispondenti alle dimensioni spaziali (lat, lon), che corrispondono ad un tempo ed ad una elevazione fissati. Ogni *slice* può essere composta da diverse bande. Una *slice* può essere anche suddivisa in “*tiles*”, cioè in sottoelementi di dimensione prestabilita; oltre a questo una *slice* può essere costituita da una piramide, ovvero può contenere delle “*overview*”, cioè delle rappresentazioni della *slice* a risoluzioni minori, precostruite solo per fini prestazionali.

4. "Time" e servizi OWS base (WMS, WFS, WCS)

Servizi WMS

Il WMS 1.1.1 è capace di esporre dimensioni addizionali rispetto alla X e Y o latitudine e longitudine; in particolare due di esse hanno i nomi prefissati *TIME* e *ELEVATION*. Esse sono pubblicizzate nel documento di *capabilities* e possono essere presenti tra i parametri di richiesta della GetMap per specificare un criterio di selezione sui dati.

Servizi WFS

Le specifiche WFS 1.0 e 1.1 non tengono in considerazione per nulla la dimensione temporale; con il WFS 2.0 e le relative specifiche per la definizione di filtri (FE 2.0) il tempo viene supportato con queste funzionalità:

- le FE vengono integrate con i “*temporal filter*”, ovvero con gli operatori corrispondenti alle 13 relazioni spaziali individuate da Allen.
- Supporto alle *join* temporali.

Dall’analisi dei requisiti non è emersa l’esigenza di arrivare a questo livello di funzionalità, non comprendendo nessun caso d’uso reale l’utilizzo di questi operatori o di *join*.

La scelta è stata quindi quella di rimanere alla versione 1.1, ma creare un profilo applicativo per il tipo di dati che si deve servire.

Altra scelta importante per i servizi WFS è stata quella di non servire *feature* complesse per evitare di richiedere implementazioni particolari sui client. Per questa ragione, il client dovrà conoscere a priori la semantica dei dati serviti, non derivabile completamente dal GML delle *feature*. Client al di fuori della *community* di Briseide, inconsapevoli quindi del significato particolare di alcuni attributi o delle relazioni che intercorrono tra diverse *feature type*, non vedranno compromesso il loro funzionamento, ma considereranno come normali *feature type* gli oggetti temporali serviti da Briseide. Utilizzando la possibilità che lo standard offre di estendere le “*capabilities*” del servizio, il WFS di Briseide pubblicherà le *feature* temporali servite attribuendole ad una delle 4 categorie individuate. Gli schemi dati delle *feature* conterranno delle convenzioni, ovvero ai nomi di alcuni attributi sarà legato un particolare significato; in altre parole si implementerà un profilo applicativo specifico del WFS, cioè una specializzazione del WFS che fornirà funzionalità aggiuntive per determinate tipologie di *feature*.

Servizi WCS

Similmente al WMS, il WCS supporta richieste che includono i parametri *TIME* ed *ELEVATION* (rispetto al WMS, c’è una piccola differenza che riguarda la gestione della *ELEVATION*, trascurabile in questo contesto). Quindi il WCS sarà capace di:

- Estrarre i valori *TIME* ed *ELEVATION* dal modello di dati del raster multidimensionale.
- Pubblicizzare i domini relativi alle due dimensioni nel documento di *capabilities*;
- Rispondere a richieste che coinvolgono solamente un valore di *TIME* e di *ELEVATION*.

E’ meglio sottolineare che al momento la risposta ad una GetCoverage non può essere un dato multidimensionale.

5. Ingestion

L'infrastruttura realizzata da Briseide permette l'ingestione automatica di dati geografici con dimensione temporale (vedi paragrafi precedenti).

I processi di *ingestion* consentono di catalogare automaticamente (ed in tempo quasi-real time) dati vettoriali e raster sia in GeoServer che i rispettivi metadati in GeoNetwork (service catalog Briseide).

Questo significa che l'infrastruttura rispetta una serie di requisiti per permettere la catalogazione "event-driven", come per esempio l'arrivo di file su un sito specifico FTP.

Per questa ragione i Servizi ingestione (IS) sono una componente essenziale in quanto rappresentano il punto di ingresso principale per l'ingestione dei dati. Inoltre, essi sono responsabili per la corretta archiviazione e catalogazione. I processi sono in grado di riconoscere anche la variabile "time", catalogando di conseguenza i layers nel wms-t di geoserver.

Il componente software scelto per realizzare i servizi di *ingestion* è GeoBatch.

GeoBatch è un progetto *open source*, basato su "eventi" per l'elaborazione batch automatica di dati geografici.

GeoBatch fornisce componenti di base per la raccolta, elaborazione e pubblicazione dei dati ad applicazioni esterne come GeoServer (<http://docs.codehaus.org/display/GEOBATCH/Geospatial+Batch+Engine>).

Un processo batch in GeoBatch è un file di configurazione XML denominato "flusso". Ogni flusso è costituito da tre sezioni: una parte descrittiva, una seconda dedicata ai dati di monitoraggio flussi (eventGeneratorConfiguration) e quello finale che è dedicato al riconoscimento di determinati file all'interno di un flusso, alla sua elaborazione e pubblicazione finale (eventConsumerConfiguration).

Nel progetto Briseide sono stati definiti 4 flussi di *ingestion*:

- Raster singola
- Imagemosaic
- ESRI Grid
- Shapefile

I quattro flussi possono essere attivati all'interno di ogni *pilot*, in base alla reale necessità e il formato dei dati forniti.

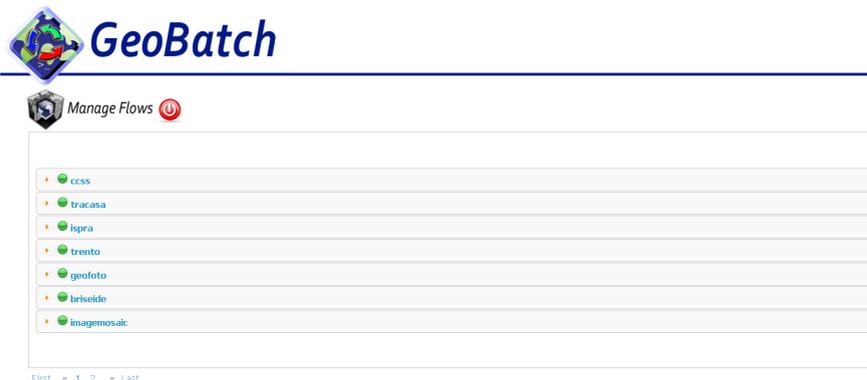


Figura 2 – Interfaccia amministrazione flussi di ingestione.

L'applicazione web di amministrazione di GeoBatch è disponibile all'indirizzo <http://briseide03.ingr.briseide.eu:8080/geobatch>: questo permette agli amministratori dei flussi di *ingestion* di monitorare i singoli flussi:

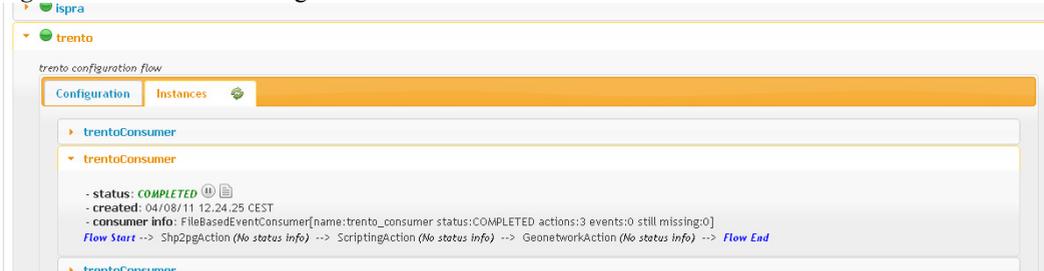


Figura 3 – Monitoraggio flusso di ingestion.

6. Processing

I servizi WPS sviluppati in Briseide sono stati implementati secondo la versione 1.0.0 , sono suddivisi in 7 tipologie:

- *topological* (*buffer, union, intersection, inclusion,*)
- *statistical* (*media, mediana, minimo, massimo, ...*)
- *transformation* (*coordinate, formato*)
- *change detection* (*differenze tra coppie di layer, sia raster che vettoriali*)
- *feature extraction* (*estrazione feature vettoriali da raster*)
- *raster algebra* (*riclassificazione e operazioni tra raster*)
- *statistic-time driven*

I primi sei WPS sono parte del progetto *open source* GeoServer: i primi tre servizi sono stati inclusi nella versione 2.1.1 rilasciata a giugno 2011 (<http://geoserver.org/display/GEOS/GeoServer+2.1.1>). Questo è un esempio concreto di contributo del progetto Briseide alla comunità *open source*.

Il settimo WPS, invece, è stato implementato utilizzando la soluzione R2WPS.

I servizi WPS sviluppati su GeoServer sono disponibili all'indirizzo <http://briseide02.ingr.briseide.eu/8080/geoserver>; è possibile testare i 6 WPS tramite la pagina di Request Builder disponibile all'indirizzo <http://briseide02.ingr.briseide.eu:8080/geoserver/web/?jsessionid=E02958C9CEDB526C7D994B7243D2C821?wicket:bookmarkablePage=:org.geoserver.wps.web.WPSRequestBuilder>, selezionando il processo che si vuole utilizzare e inserendo i parametri di *input*.

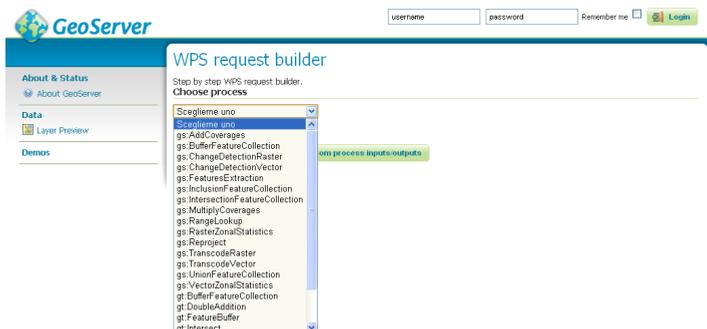


Figura 4 – Elenco WPS.

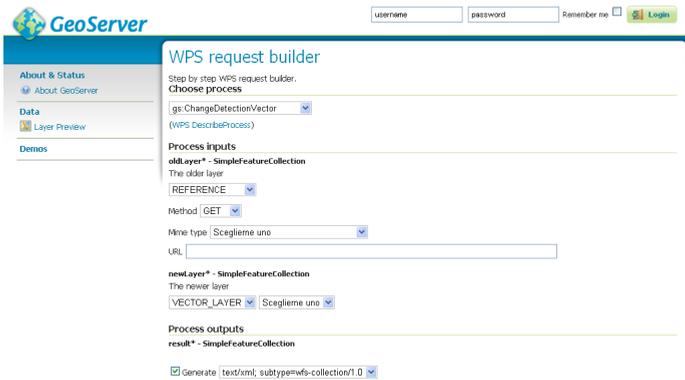


Figura 5 – Interfaccia test WPS.

Nell’immagine seguente c’è un semplice esempio di un’operazione di *buffer* (WPS “topological”): il *client* (gvSIG 1.9) utilizza un servizio WFS esterno e su una *feature* del *layer* “parks” definisce un *buffer* di 1000 metri, chiedendo come output al servizio WPS un *layer* di *buffer* in formato SHP. Inoltre è possibile concatenare il processi WPS in modo tale che l’*output* di un processo diventi l’*input* di un’altro processo (si pensi ad esempio ad una operazione di *buffer* su un *layer* ottenuto come intersezione o unione di altri due *layer*).

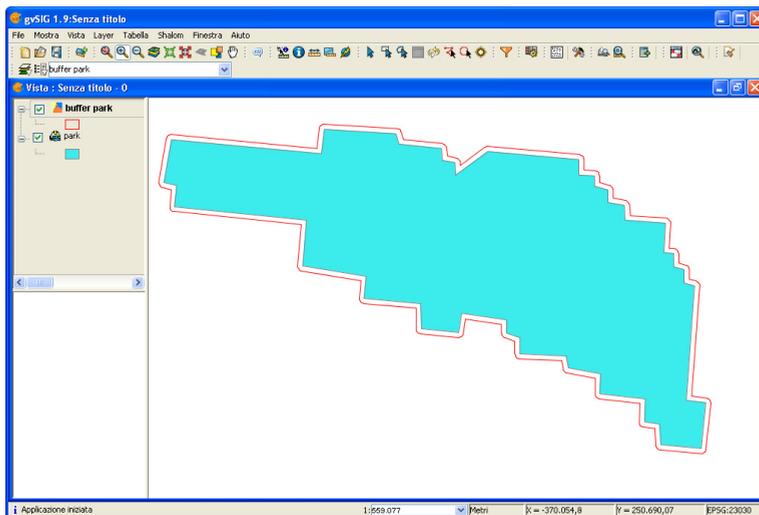


Figura 6 – Visualizzazione risultato buffer.