Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Corso di laurea triennale in Ingegneria Informatica

VESTIZIONE DI DATABASE TOPOGRAFICI IGM UTILIZZANDO SLD

RELATORE:

PROF. MASSIMO RUMOR

CORRELATORE:

DOTT. SANDRO SAVINO

LAUREANDO:

NICCOLÒ PRETTO - 561915 IF

ANNO ACCADEMICO:

2009/2010

University of Padova - Italy

GIS RT labs

GEOGRAPHICAL INFORMATION SYSTEMS
REAL-TIME SYSTEMS

DEPARTMENT OF INFORMATION ENGINEERING

UNIVERSITY OF PADOVA

Ai miei genitori e alle mie sorelle per il sostegno che mi danno ogni giorno

Agli amici e ai compagni per tutti i momenti passati insieme

Ad Emma per la gioia che mi regala ad ogni suo sorriso

Sommario

1 INTRODUZIONE 2
1.1 CARTOGRAFIA E GENERALIZZAZIONE CARTOGRAFICA 2
1.1.1 CENNI STORICI 2
1.1.2 LA CARTA GEOGRAFICA E LE VARIE TIPOLOGIE 3
1.1.3 PROCESSO CARTOGRAFICO 4
1.1.4 GENERALIZZAZIONE 6
1.2 GIS E CARTOGRAFIA DIGITALE 8
1.2.1 GIS 8
1.2.2 CARTOGRAFIA DIGITALE 9
1.3 PROGETTO CARGEN 12
1.3.1 IL PROGETTO 12
1.3.2 FASI DEL PROGETTO 12
2 VESTIZIONE 13
2.1 INTRODUZIONE ALLA VESTIZIONE 13
2.2 SLD 15
3 PROGETTO 17
3.1 INTRODUZIONE 17
3.1.1 OBIETTIVI 17
3.1.2 PERCORSO 18
3.1.3 TECNOLOGIA UTILIZZATA 19
3.2 FOGLIO DI STILE 20
3.2.1 ATTRIBUTI ASSOCIATI AL CODICE LAB DI UNA FEATURE 21
3.2.2 POINT SYMBOLIZER 25
3.2.3 LINE SYMBOLIZER 27
3.2.4 POLYGON SYMBOLIZER 27
3.3 PARSER 28
3.4 COMPLEX STYLE 28
3.5 PLUGIN 30
3.5.1 SINGLE LAYER 31
3.5.2 DB LAYERS 32
4 CONCLUSIONI 35
4.1 RISULTATI 35
4.2 PRESTAZIONI 36
4.3 VANTAGGI 36
4.4 SVILUPPI FUTURI 37
5 ELENCO FIGURE 39

1 INTRODUZIONE

Questo lavoro si sviluppa nell'ambito del progetto di ricerca CARGEN, svolto all'interno del laboratorio GIRST (Geographical Information Systems Real Time Systems) del Dipartimento di Ingegneria dell'Informazione dell'Università degli Studi di Padova.

La mia esperienza all'interno del progetto è incentrata sulla realizzazione di un plug-in per il software GIS OpenJump che permetta di visualizzare i dati provenienti da database topografici in scala 1:25000 e 1:50000 utilizzando la vestizione standard sviluppata dall'IGM (Istituto Geografico Militare).

Nei paragrafi successivi a questa breve introduzione verranno introdotti i concetti basilari della cartografia, i fondamenti e gli strumenti della cartografia digitale e della generalizzazione e verrà descritto il progetto CARGEN, per permettere al lettore una più approfondita comprensione del lavoro svolto, degli strumenti e degli standard utilizzati e dei risultati ottenuti con la realizzazione del plug-in.

1.1 CARTOGRAFIA E GENERALIZZAZIONE CARTOGRAFICA

Una carta geografica è una rappresentazione grafica, riprodotta in scala e secondo simboli convenzionali, di una parte o di tutta la superficie terrestre, o di alcuni suoi aspetti particolari

(Dizionario Garzanti)

1.1.1 CENNI STORICI

Gli storici della cartografia hanno fornito diverse versioni su quelli che possono essere considerati i documenti cartografici più antichi a noi pervenuti. Il più antico reperto sembra essere, per ora, un graffito risalente a circa 15.000 anni fa, ritrovato a Mezin, in Ucraina, che rappresenta un fiume che tuttora scorre nella zona del ritrovamento, e un villaggio che sorge sulle sue sponde. Possono essere citati molti altri esempi di primitiva cartografia, come la figura 1, la tavoletta di argilla raffigurata nella figura affianco, che risale a un periodo compreso tra il 2300 e il 2500 a.C. ; inoltre, alcuni scavi in Anatolia Centrale hanno portato alla luce una rappresentazione rupestre che indica una mappa raffigurante un insieme di abitazioni (circa 80), del 6200 a.C. .

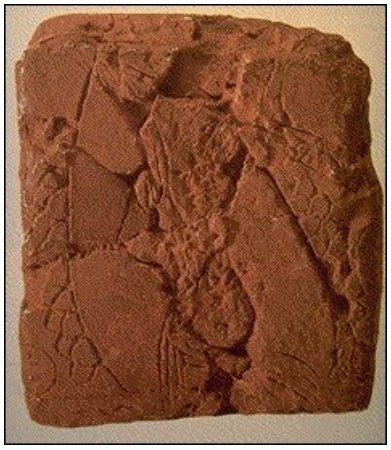

Figura 1 : Tavoletta con pianta di distretto agricolo, da Nuzi, Mesopotamia

Ciò che appare evidente da queste testimonianze, è che l'uomo fin dai tempi più antichi ha ritenuto utile rappresentare graficamente il mondo che lo circondava, per conoscere e comprendere il proprio territorio.

Queste prime forme cartografiche raccoglievano informazioni semplici, riguardanti case, fiumi e poche altri elementi geografici. Di pari passo con l'evoluzione delle varie civiltà, il mondo da descrivere è divenuto sempre più complicato da rappresentare. Per raffigurare tutte le informazioni necessarie, è sorta la necessità di sviluppare e affinare le tecniche topografiche.

Per assistere a un forte sviluppo della cartografia bisogna attendere il Rinascimento, con la riscoperta dell'opera di Tolomeo, l'introduzione dei primi testi tradotti dall'arabo e con le grandi imprese dei navigatori. La creazione dei primi strumenti per l'osservazione celeste e terrestre, il perfezionamento della bussola e degli orologi, e l'invenzione della stampa, permettono la creazione e la diffusione di mappe sempre più accurate.

I primi istituti Cartografici Nazionali sorgono già nel 1791 in Francia e nel 1817 in Inghilterra. I primi a dotarsi di una cartografia nazionale furono i francesi nel 1818, mentre quella italiana arrivo più di un secolo dopo, nel 1921.

Gli ultimi anni del secolo scorso hanno conosciuto un nuovo grande momento di sviluppo della cartografia, grazie all'introduzione di strumenti informatici e alla nascita dei primi Sistemi Informativi Geografici. La staticità del tradizionale supporto cartaceo, è stata sostituita da una visualizzazione dinamica dei dati tramite monitor, creando nuove potenzialità di utilizzo e sviluppo e, contemporaneamente, introducendo nuove problematiche e necessità.

I GIS, da poco più di un'alternativa nella visualizzazione dei dati, diventano potenti strumenti di analisi dei dati geografici: le informazioni e le relazioni spaziali possono venire elaborate, manipolate, modificate. Le mappe, per essere utilizzate con questi nuovi strumenti, vengono dapprima digitalizzate e successivamente iniziano a venire prodotte unicamente in formato digitale: si assiste alla nascita delle carte numeriche, evolutesi poi in database geografici (GeoDB). GIS e GeoDB rivoluzionano il modo di trattare e di concepire il dato geografico: gli oggetti spaziali non sono più solo una collezione di segni grafici e simboli, ma sono entità dotate di attributi, inserite in gerarchie relazionali, cui corrisponde non solo una rappresentazione grafica, ma anche un contenuto logico e semantico. Il GeoDB non memorizza solo un disegno semplificato di un certo territorio, ma un modello concettuale coerente di una certa realtà, e il GIS è lo strumento tramite cui interagire con le informazioni veicolate in tale modello.

1.1.2 LA CARTA GEOGRAFICA E LE VARIE TIPOLOGIE

*La **cartografia** è l'insieme di conoscenze scientifiche, tecniche e artistiche finalizzate alla rappresentazione simbolica ma veritiera di informazioni geografiche - o statistiche, demografiche, economiche, politiche, culturali, comunque in relazione al luogo geografico nel quale si realizzano - su supporti piani (carte geografiche) o sferici (globi).* (Wikipedia)

Le carte geografiche vengono comunemente suddivise in due distinte tipologie:

- *Carte topografiche* che forniscono informazioni metriche e descrittive di interesse generale della superficie fisica della Terra o di una porzione di questa. Oltre a informazioni

orografiche e idrografiche, sono presenti anche le opere umane. Ad esempio possono essere indicate le principali vie di trasporto, o i principali insediamenti umani, sullo sfondo dello spazio naturale (fiumi, coste, rilievi, etc.)

- *Carte tematiche* che forniscono informazioni specifiche su circoscritti fenomeni di particolare interesse.

Un aspetto di basilare importanza nella cartografia, e più precisamente nella definizione di una mappa, è il concetto di scala di rappresentazione (o semplicemente scala). Può essere definita come *il rapporto tra le dimensioni della realtà e quella di una sua rappresentazione*. (Wikipedia)

Solitamente viene indicata con una frazione avente per numeratore l'unità e per denominatore il numero che indica quante volte bisogna moltiplicare una lunghezza misurata sulla carta per ottenere la corrispondente misura reale (es. 1:10000 o 1/10000). Ad esempio su una mappa in scala 1:25000 lo spessore di un tratteggio di 1 mm corrisponde a 25 metri nella realtà.

Risulta chiaro come la scala di rappresentazione influisca direttamente sulla precisione della carta: assumendo, ad esempio, che l'occhio umano riesca a percepire linee di spessore minimo pari a 0,1 mm, riportandosi all'esempio precedente di una carta in scala 1:25000 appare evidente che nonostante l'accuratezza della carta, essa sarà soggetta a un errore di circa 2,5 metri.

Le carte sono classificate in base alla scala:

- *Carte a grandissima scala* (1:2000 – 1.1000)
- *Carte a grande scala*, piante o mappe fino a 1:10.000. Piante di città, mappe, Carte Tecniche Regionali, carte catastali.
- *Carte topografiche*, in scala 1:10.000 - 1:150.000. Carta Tecnica Regionale (CTR) 1:10000; carte a grande scala dell' IGM, con scale 1:50000, 1:100000.
- *Carte corografiche* o regionali, in scala 1:150.000 - 1:1.000.000
- *Carte a piccola scala*, generali o geografiche, con scale superiori a 1:1.000.000.

1.1.3 PROCESSO CARTOGRAFICO

Con processo cartografico si definisce tutto l'insieme di procedimenti e operazioni necessari alla creazione di una carta geografica.

È rimasto sostanzialmente invariato nel tempo, e può essere schematizzato nei passi seguenti:

1. *Definizione*
2. *Analisi*
3. *Raccolta dei dati*
4. *Costruzione della mappa*
5. *Collaudo*

Definizione e Analisi

Nel processo di definizione e analisi si decidono le caratteristiche che la mappa dovrà possedere. I parametri da definire sono molteplici: tra questi si possono citare la scala, la proiezione e il supporto fisico su cui verranno rappresentati i dati.

In queste fasi viene stabilito cosa la mappa deve rappresentare e in quale modo deve farlo. Le scelte fatte influenzano sia caratteristiche tecniche che semantiche, ovvero relative ai contenuti. Per le prime vengono scelti parametri quali superficie di riferimento, proiezione adottata, la scala, etc. . Per le altre le scelte variano a seconda che si voglia focalizzare l'attenzione su particolari aspetti della realtà (cartografia meristica o tematica) o si ricerchi di ottenere una rappresentazione più completa possibile (cartografia olistica o analitica).

Le decisioni prese in questa fase definiscono una prima astrazione della realtà e delineano a livello semantico un primo modello della realtà. Questa modellazione, inoltre, nella cartografia tradizionale su supporto cartaceo porta alla creazione della legenda, mentre in quella digitale alla definizione del geodatabase.

Acquisizione dei dati

L'acquisizione dei dati differisce a seconda della provenienza dei dati di origine. Se questi sono ottenuti direttamente dalla realtà, tramite acquisizione diretta sul territorio, si parla di *mappe rilevate*. Nel caso in cui le informazioni siano estratte da cartografia preesistente le mappe si dicono *derivate.* È importante specificare che queste ultime si possono ottenere solo utilizzando carte a scala maggiore rispetto a quella della mappa che si vuole ottenere.

La tecnica più usata per l'acquisizione diretta delle informazioni è la fotogrammetria. Questa prevede l'utilizzo di immagini fotografiche stereoscopiche del terreno. Solitamente queste immagini sono riprese da un aereo, in sequenze denominate strisce (strip). Ogni striscia si sovrappone con quelle adiacenti di circa il 15% su entrambi i lati. Appositi strumenti, detti stereorestitutori, permettono di associare le coordinate degli oggetti contenuti nelle strisciate e restituirne latitudine, longitudine ed elevazione.

Durante questa fase le informazioni vengono interpretate e classificate: ad ogni oggetto acquisito viene assegnato un codice, che stabilisce a quale specifico elemento del modello precedentemente creato questo appartiene (es. casa, ferrovia, autostrada, fiume, etc.).

Costruzione della mappa

La fase successiva all'acquisizione dei dati è la costruzione della mappa. Il cartografo deve produrre una mappa che non solo soddisfi le specifiche delineate nella fase di definizione e analisi, ma che risulti usabile e leggibile. Per svolgere il lavoro non esistono però schemi precisi, ma si deve affidare alla sua conoscenza ed esperienza. Un altro suo compito consiste nell'estrarre il contenuto dei dati di partenza creando un'astrazione della realtà efficace e rappresentativa che faciliti la

comprensione dell'informazione. Questo processo è definito *generalizzazione cartografica,* e verrà descritto in maniera più completa nel prossimo paragrafo.

La fase di costruzione viene finalizzata con un raffinamento estetico. Possono essere citati vari esempi quali la creazione di ombreggiature per il delineamento dell'orografia, il posizionamento di riferimenti, l'assegnazioni di vestizioni ai vari particolari.

Collaudo

Il collaudo consiste in una serie di test mirati alla verifica di correttezza e consistenza della carta prodotta. Questa fase può produrre anche un raffinamento estetico del prodotto. Alcuni esempi di test sono il controllo dei punti d'appoggio, della restituzione, del disegno e il controllo finale sul terreno mediante operazioni di misura e verifica della rappresentazione cartografica, che si svolge confrontando i dati rilevati sul terreno tramite strumenti ad alta precisione (per esempio GPS differenziale) con i dati riportati sulla carta.

1.1.4 GENERALIZZAZIONE

"*Generalization aims to provide an abstraction of geographic reality to enhance comprehension and communication of information"*

(Agent Esprit, 2001)

La generalizzazione cartografica è il processo che filtra e semplifica i vari elementi della realtà geografica, creando un modello della realtà per poter ottenere una rappresentazione che soddisfi la necessità informativa per cui la mappa è stata creata. Sono necessarie, quindi, opportune scelte spaziali per poter bilanciare accuratezza e contenuti, per ottenere così una buona leggibilità della mappa alla scala desiderata.

Figura 2: Esempio di mappa generalizzata a varie scale

Solitamente la generalizzazione viene suddivisa in due tipologie che si differenziano per la base informativa su cui viene svolto il processo. Se la creazione della mappa è impostata su dati cartografici preesistenti si parla di *map generalization*. Nel caso in cui si utilizzino dati ottenuti tramite acquisizione diretta (per esempio con scatti di ripresa fotogrammetrica) il processo viene definito *map compilation*. Nonostante questa distinzione, entrambe le tipologie devono risolvere gli stessi problemi relativamente al contenuto della mappa e alla sua rappresentazione; devono, inoltre, sottostare alle specifiche della carta e ai parametri che la contraddistinguono. La differenza fondamentale risiede

nel diverso lavoro per ottenere il modello astratto dei dati, diverso se i dati di partenza sono derivato o rilevati.

Il processo di generalizzazione può essere diviso essenzialmente in due fasi successive:

- **Model-oriented generalization:** ha l'obiettivo di creare un modello astratto che rappresenti la realtà di interesse. In questa fase vengono valutati i vari aspetti della realtà, che verranno mantenuti, se realmente necessari allo scopo della mappa specifica, o eliminati nel caso in cui il loro contenuto informativo risulti superfluo. Il passo successivo è la classificazione dei vari elementi per ottenere un primo livello di astrazione. I parametri che portano alla suddivisione degli elementi in varie classi differiscono a seconda dello scopo per cui la mappa è stata commissionata, e vengono decisi nella fase di definizione e analisi. Alla realizzazione del modello viene poi affiancata la creazione di un GeoDB ovvero un database che contiene i dati precedentemente raccolti, filtrati e classificati. Questo inizialmente verrà popolato con i dati e le classi tassonomiche più rilevanti e, solo successivamente, verrà ampliato con informazioni maggiormente strutturate e organizzate per ottenere maggiore efficacia ed efficienza nelle successive procedure di analisi dei dati.

- **Cartographic generalization:** consiste nella rappresentazione del modello dati precedentemente trovato. Questa operazione è estremamente complessa a causa della sua arbitrarietà: il cartografo, infatti, si deve spesso affidare all'interpretazione e all'esperienza personale per raffigurare simbolicamente i dati; non esistono, infatti, schemi rigidi da seguire. In questa fase viene introdotta un'ulteriore astrazione della realtà dovuta all'aggiunta di approssimazioni ed errori inseriti dall'addetto per migliorare la densità dei contenuti, la leggibilità e il contrasto visivo della carta o del supporto digitale su cui verrà disegnata la mappa.

Il processo di generalizzazione cartografica è solitamente molto lento e costoso, e incide molto su tempi e spese per la produzione della mappa. Da qui nasce la necessità di automatizzare il lavoro per renderlo più rapido ed efficiente, senza però penalizzare la qualità della generalizzazione poiché condiziona direttamente la qualità e l'utilità del prodotto cartografico finale.

1.2 GIS E CARTOGRAFIA DIGITALE

1.2.1 GIS

Un *Geographic Information System* (GIS) è un sistema informativo il cui scopo è quello di collezionare, manipolare, analizzare e visualizzare dati spaziali e alfanumerici del mondo reale, grazie alla sua capacità di mettere in relazione diverse informazioni relative allo stesso contesto spaziale. Le grandi potenzialità di questi sistemi risiedono, infatti, nella capacità di intrecciare le informazioni raccolte e relazionarle per poter ottenere soluzioni che soddisfino le necessità dell'utente circa i rapporti che intercorrono tra di esse.

La creazione di questi strumenti si basa sulla fusione di due capisaldi dell'innovazione informatica:

- I sistemi di disegno computerizzato CAD, che permettono di rappresentare graficamente gli elementi geografici;
- I database relazionali che consentono di immagazzinare le informazioni relative alle varie entità.

I database che permettono di memorizzare, interrogare e manipolare informazioni geografiche e dati spaziali sono detti **geodatabase**. Le informazioni che raccolgono e gestiscono possono essere relativi a tre tipologie:

- **Geometriche**, relative alla rappresentazione cartografica dell'oggetto (forma, posizione, dimensione, etc.).
- **Topologiche**, in riferimento alle relazioni reciproche tra gli elementi (sovrapposizione, adiacenza, etc.).
- **Informative**, attinenti ad attributi non spaziali degli oggetti in questione.

Le geometrie dei dati spaziali possono essere memorizzate come punti, linee e poligoni, e sono associati a un sistema di riferimento spaziale. Un record del geodatabase può così usare i dati geometrici per rappresentare un oggetto nel mondo reale. A questi dati spaziali vengono poi associati altri dati alfanumerici dei database standard per poter memorizzare altri attributi che li caratterizzano. Molti geodatabase possiedono funzioni personalizzate per poter manipolare e interrogare i dati tramite SQL, per esempio per trovare tutti i punti di un determinato tipo contenute in una determinata area.

Grazie ai sistemi GIS le applicazioni della cartografia si moltiplicano: non solo è possibile rappresentare in una carta ogni dato grazie alle sue coordinate e risalire a tutte le informazioni che lo riguardano, ma è anche possibile analizzare le proprietà geometriche delle entità rappresentate (ad esempio le loro esatte dimensioni fisiche) e combinarle con le proprietà generali delle altre entità cartografiche (ad esempio la distanza reciproca tra case, abitazioni, tra scuole e ospedali, etc.) e con tutte le informazioni che riguardano queste ultime.

1.2.2 CARTOGRAFIA DIGITALE

Grazie all'ausilio di sistemi informatici e dei GIS, negli ultimi trent'anni la situazione della cartografia è profondamente cambiata. Innanzitutto le carte vengono ora realizzate sulla base delle informazioni contenute in un database: più precisamente, sono dei sottoprodotti del database. Il computer non viene più utilizzato solamente per automatizzare le tecniche descrittive dei cartografi, ma è diventato lo strumento di valutazione della qualità dei dati, l'elemento che fonde i dati tra loro e ne valuta la compatibilità; viene utilizzato sia dal ricercatore che cerca fonti e materiali interessanti ai fini del lavoro, sia dal cartografo e dal grafico, che riproducono i dati raccolti.

Nei paragrafi successivi verranno presentati la situazione alcuni esempi di carte digitale e la situazione cartografica italiana.

CARTA D' ITALIA IN SCALA 1:50000

La Carta d'Italia in scala 1:50000 IGM è composta dai 652 fogli della serie 50, ed è derivata dai dati acquisiti per la formazione delle tavolette della serie 25DB. I fogli della serie 50 IGM non vengono disegnati a partire da un database topografico. Le specifiche e le informazioni necessarie per la formazione della Carta d'Italia sono contenute in un unico documento, "*Norme e Segni Convenzionali per la realizzazione dei fogli della Carta d'Italia alla scala 1:50000*", edito da l'Istituto Geografico Militare nel 2004, che fornisce principalmente l'elenco degli oggetti topografici, i segni convenzionali, cioè la descrizione del simbolismo ad essi associato, alcune specifiche relative all'acquisizione degli stessi ed alcune indicazioni relative alle operazioni di tali elementi .

La Carta contiene 444 particolari topografici distinti divisi in 11 gruppi, ognuno dei quali è diviso a sua volta in sottogruppi. I gruppi presenti sono i seguenti:

- Vie di comunicazione ferrate
- Viabilità
- Elementi divisori
- Insediamenti
- Costruzioni e particolari
- Impianti
- Particolari idrografici
- Vegetazione
- Limiti amministrativi
- Orografia
- Riferimenti planimetrici

Ogni simbolo è numerato, con numerazione progressiva all'interno di ogni gruppo, da 201 a 1208. Tra i simboli, la loro numerazione e le definizioni non esiste una relazione strettamente univoca: alcuni simboli sono associati a più definizioni (es. il simbolo 203, usato sia per tranvie in sede propria, sia per ferrovie a scartamento ridotto e funicolari/cremagliere), alcuni numeri

corrispondono a più simboli (è il caso ad esempio del simbolo 509 per moschee e sinagoghe), e alcune definizioni corrispondono a più simboli (es. la definizione di centrali elettriche).

DB25

Il modello dati utilizzato è il DB25, creato da IGM per la compilazione della cartografia 25DB. Questo modello è caratterizzato da una rappresentazione del mondo reale attraverso l'utilizzo di features identificate attraverso un attributo LAB (Label). Ogni feature appartiene inoltre a una categoria identificata con un codice FACC (Feature and Attribute Coding Catalugue).

Ogni feature è caratterizzata da una tipologia di primitiva grafica evidenziata dalla prima lettera del codice:

- C per la tipologia areale
- L per quella lineare
- P per quella puntuale
- T per le feature di tipo testo

Gli oggetti topografici del DB25 sono divisi in strati informativi o layers.

Le features del DB25 non presentano che una minima organizzazione gerarchica: ogni oggetto topografico è di norma descritto direttamente da un codice LAB univoco; questa scelta si spiega nella finalità tipografica di questo modello di dati: a ogni codice LAB è, infatti, associata anche una vestizione grafica da usare poi nella stampa delle carte e riportare nella legenda delle cartografie in serie 25DB.

Il DB25, che nella versione originale IGM precedente alla revisione del progetto CARGEN, conteneva 291 particolari topografici, dopo la revisione ne contiene 226, divisi nei seguenti strati:

- Altimetria
- Confini
- Geomorfologia
- Idrografia
- Industrie
- Insediamenti
- Servizi
- Toponomastica
- Trasporti
- Vegetazione

IGM E SITUAZIONE CARTOGRAFICA ITALIANA

Per poter comprendere l'importanza strategica e il valore reale delle ricerche che si svolgono all'interno del laboratorio GIRST è necessaria una breve presentazione sulla situazione attuale relativa alla produzione cartografica da parte dell'IGM e dei vari organi regionali predisposti a tale scopo.

L'Istituto Geografico Militare, fondato nel 1872, è l'organismo predisposto al mantenimento di una cartografia a livello nazionale. L'istituto ha creato la prima mappa nazionale (Nuova Carta Topografica d'Italia) in scala a 1:100000, formata da 278 elementi detti "fogli" con dimensione di 30' in longitudine e 20' di latitudine e ha impegnato l'IGM dal 1875 al 1921 anche se con alcune pause.

Attualmente l'IGM mantiene più carte del territorio nazionale realizzate in diverse scale. Per la scala 1:25000 esistono le serie 25, 25V e 25DB, per quella a 1:50000 sono disponibili le serie 50 e 50L, mentre per quella a 1:100000 le serie 100V e 100L.

Le serie a scala 1:25000 sono le carte di scala maggiore mantenute dall'istituto, e quindi con maggior dettaglio e con tempi di produzione più lunghi. Per comprendere appieno l'enorme mole di lavoro, occorre considerare che la prima mappa realizzata in questa scala era composta da 3545 elementi denominati "tavolette". Al momento solo le tavole della serie 25V, che riferiscono alla vecchia produzione, sono state completate. Della nuova serie 25, invece, sono stati prodotti solo 840 elementi dette "sezioni" rispetto alle 2298 previste. Questo significa che la copertura cartografica è affidata mediamente a carte risalenti mediamente al 1960, con l'eccezione di alcune prodotte in una campagna di aggiornamento del 1984. La serie 25DB è tutt'ora in fase di produzione.

A seguito dell'approvazione del DPR no616 del 24 luglio 1977 è avvenuto il "Trasferimento delle funzioni alle regioni in materia di ambiente e territorio", grazie al quale le regioni possono gestire autonomamente la creazione delle mappe regionali.

In Veneto vengono prodotte due tipologie di carte tecniche regionali denominate CTR: una in scala 1:5000 e una in scala 1:10000, il cui taglio le rende sovrapponibili con le carte IGM di nuova produzione (non quindi a quelle di serie 25V).

La produzione di mappe fino ad ora è stata possibile attraverso rilievi aerofotogrammetrici, oppure con tramite rilievi aerofotogrammetrici numerici o analogici seguita da una fase di rappresentazione grafica eseguita con metodologie automatiche o manuali. La situazione sta, però, velocemente cambiando da quando IGM e Regioni hanno intrapreso un processo di digitalizzazione delle relative cartografie. La disponibilità di questi dati digitali permette di ottenere le sezioni della serie 25DB non solo attraverso stereo restituzione numerica, ma anche mediante la derivazione dalla Cartografia Tecnica Regionale Numerica.

Già da parecchi anni si è assistito a una sempre più rapida conversione della produzione cartografica a sistemi digitali; recentemente questa tendenza sta finalmente convergendo verso la creazione non più di semplice cartografia digitale, bensì di più moderni geodatabase. Recentemente, in questa prospettiva, è stato approvato dal CNIPA un documento, che presto diventerà legge, che stabilisce un modello dati nazionale per i geodatabase ad alta scala da realizzare ed utilizzare nelle amministrazioni pubbliche. La disponibilità di dati geografici ad alta scala e in formato digitale, rende ipotizzabile la realizzazione di una procedura automatica per la generalizzazione cartografica permetterebbe non solo di accelerare i tempi di produzione delle mappe, ma soprattutto di raggiungere un obiettivo molto più ambizioso: la possibilità di derivare in maniera completamente

automatica le carte 25DB dalla cartografia a grande dettaglio delle cartografie regionali. Proprio in questo ambito è nato e si sta sviluppando il progetto CARGEN.

1.3 PROGETTO CARGEN

1.3.1 IL PROGETTO

CARGEN (CARtographic GENeralization) è un progetto nato nel 2007 da una convenzione tra Regione Veneto e il Dipartimento di Ingegneria dell'Informazione dell'Università di Padova, con la collaborazione dell' Istituto Geografico Militare Italiano, che coinvolge studenti e ricercatori nell'attività di ricerca sulla generalizzazione cartografica.

Il primo obiettivo del progetto era di individuare le metodologie informatiche di derivazione per realizzare un prototipo consistente del DB25 IGM dell'area del Parco delle Dolomiti Bellunesi, a partire dalle carte tecniche regionali numeriche DB5k. Recentemente questo obiettivo è stato ampliato includendo la creazione di un prototipo del DB50 IGM della medesima area territoriale.

L'area campione utilizzata è una porzione del territorio di circa 132.000 ettari che contiene al suo interno quasi la totalità degli elementi geografici presenti nel modello previsto dagli standard IGM, e si presta quindi perfettamente per poter testare sperimentalmente i metodi di derivazione individuati.

Per l'area campione sono stati messi a disposizione dalla Regione Veneto i seguenti dati:

- CTR in formato digitale, al massimo livello di aggiornamento (2004)
- Ortofotocarta in formato digitale alla scala 1:5.000 o superiore
- Fotogrammi delle riprese aeree relative alla cartografia fornita
- GeoDBR

A questi si aggiungono i dati del prototipo realizzato del DB25.

1.3.2 FASI DEL PROGETTO

Il progetto si comporrà delle seguenti parti:

- La definizione del modello dati per il DB_50 IGM
- La creazione di una mappatura tra le feature del DB_50 IGM e le feature di interesse presenti nei database topografici alle scale maggiori
- La realizzazione dei metodi per la derivazione
- La creazione e il popolamento del DB_50 IGM
- La produzione di cartografia a partire dal DB_50 IGM

2 VESTIZIONE

2.1 INTRODUZIONE ALLA VESTIZIONE

La qualità del design di una mappa influisce fortemente sulla sua capacità informativa. Prima dello sviluppo degli odierni geodatabase e GIS, la vestizione dei dati geografici, era in strettissimo rapporto con il contenuto informativo che una mappa offriva. In base alla colorazione e alla simbologia utilizzata la persona poteva ottenere le informazioni desiderate.

La vestizione dei dati cartografici è il processo che attribuisce a una determinata geometria (punto, linea o poligono) una specifica colorazione o simbologia in base al tipo di elemento geografico che essa rappresenta all'interno della mappa. Per creare un modello simbolico che assegni ai dati una buona vestizione, è necessario mediare l'intento di ritrarre il mondo in modo più accurato possibile e di trasmettere le informazioni nel modo più efficace possibile.

Già gli antichi romani introdussero il principio della simbologia. La *"Tabula Peutingeriana"* è una copia medioevale di una carta romana in cui si trovano più di 500 simboli diversi (città grandi o secondarie, ponti, templi, terme, punti di riposo, etc.).

Figura 3: Parte della "Tabula Peutingeriana"

Nonostante siano passati molti secoli, la vestizione ricopre tuttora un ruolo di primaria importanza.

Per ottenere una buona rappresentazione dei dati è necessario uno studio di analisi che inizia già durante la progettazione della carta. Durante la costruzione del modello, in base alla scala adottata dalla mappa, è necessario un approfondito studio sul livello del dettaglio, che influenza la quantità di informazioni contenute nella carta, e quindi la qualità e la leggibilità della rappresentazione stessa.

Nel caso in cui venga adottato un livello di dettaglio troppo elevato, infatti, c'è il rischio che i vari elementi si sovrappongano durante la loro vestizione. Un modo che per evitare ciò potrebbe consistere sulla riduzione degli spessori delle linee o più in generale delle linee, ma questo spesso porta a una perdita di informazione nelle mappe a causa di una cattiva leggibilità della stessa.

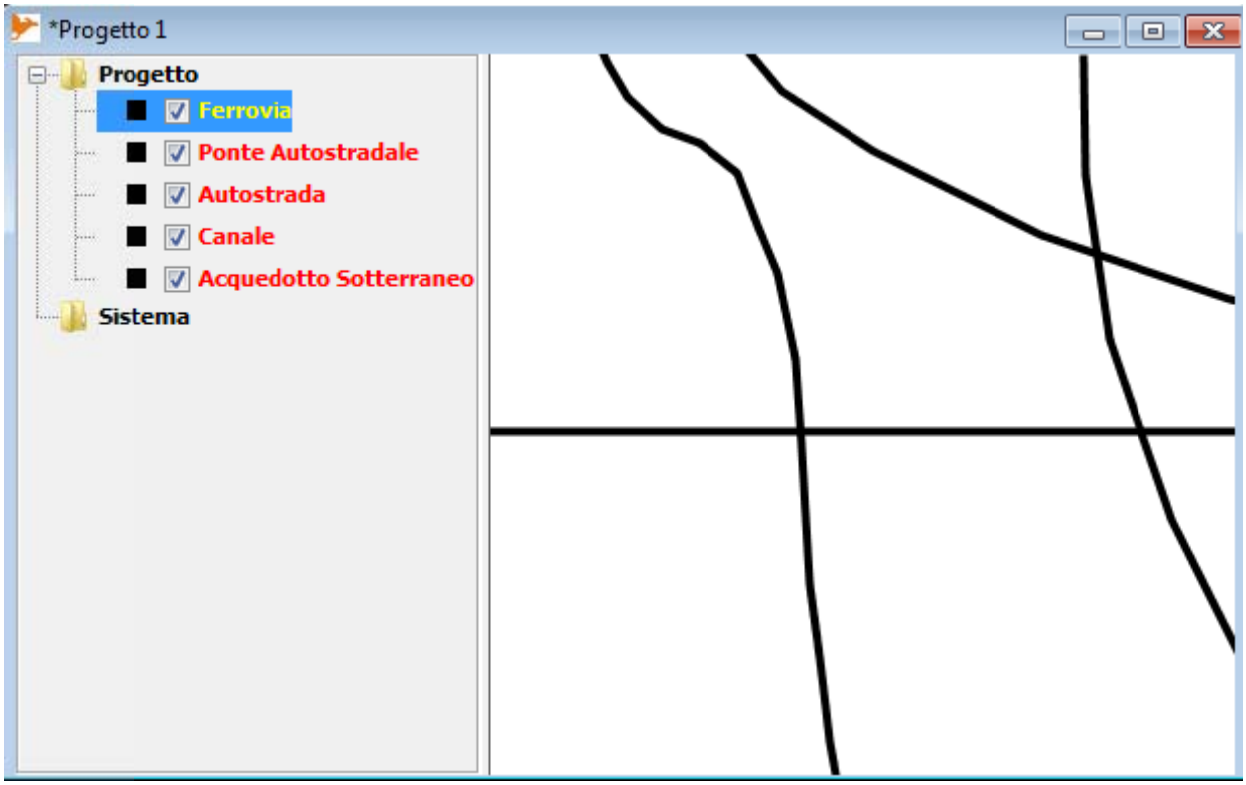

Figura 4: Porzione di carta geografica non vestita

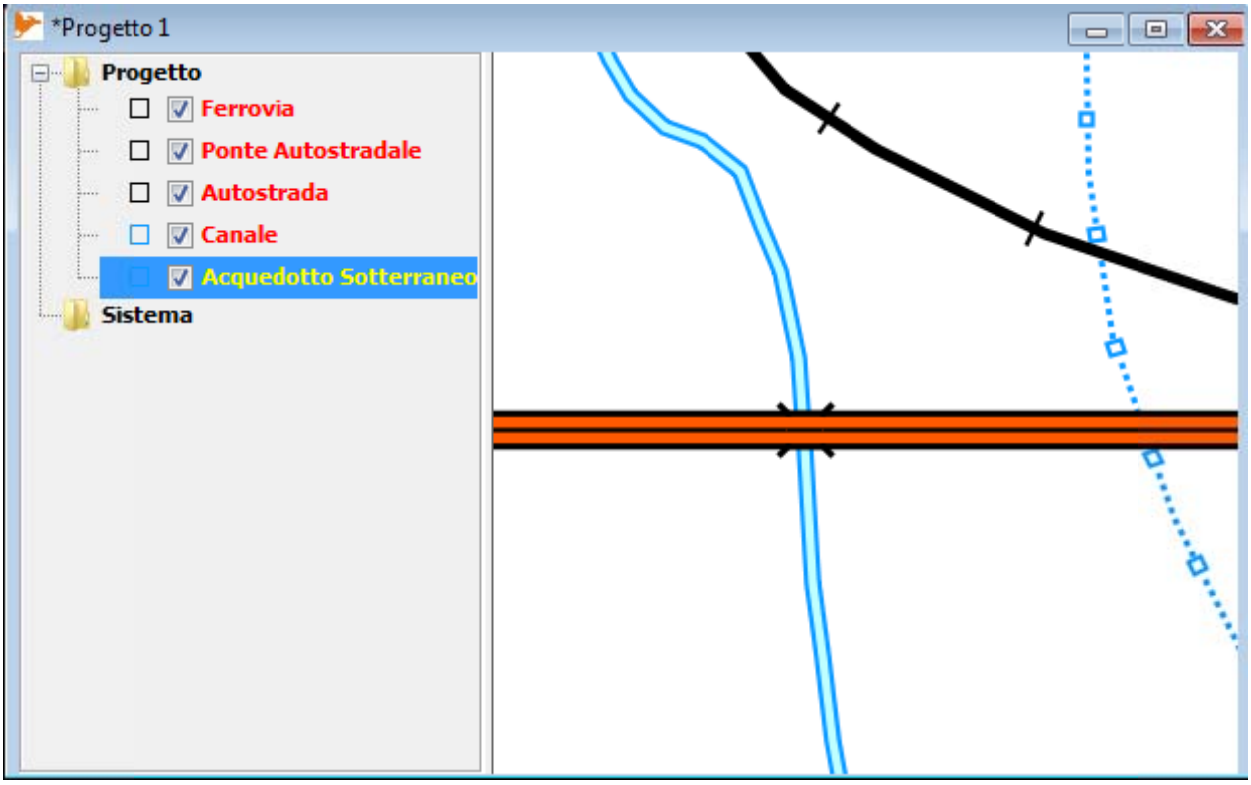

Figura 5: Porzione di carta geografica vestita

Un altro aspetto di vitale importanza per la vestizione è il colore che il cartografo associa al dato. Le varie tonalità, infatti, influiscono notevolmente sulla comprensione della mappa. Le figure 4 e 5

mostrano chiaramente come senza una vestizione, i vari elementi geografici senza colori e simbologie differenti, non forniscono nessuna informazione. La scelta delle colorazioni, in passato, era un lavoro spesso molto complicato. Da quando i personal computer sono stati introdotti nella cartografia, sono scomparsi i problemi derivati dal limitato numero dei colori utilizzabili, poiché sono visualizzabili oltre 16 milioni di colori, L'occhio umano non è in grado di distinguerli tutti, ma sono disponibili comunque abbastanza tonalità anche per le mappe più complesse. I GIS, inoltre, permettono di creare modelli e pattern che per soddisfare ogni esigenza rappresentativa.

Per costruire un buon modello per la vestizione dei dati l'OGC ha proposto un modello standard basato su XML Schema che permette la descrizione schematica di stili, e che rende esportabile questi stili sui principali GIS open source e commerciali disponibili sul mercato; lo standard verrà spiegato nel paragrafo successivo.

2.2 SLD

SLD è un XML Schema proposto da OGC (Open Geospatial Consortium) per la descrizione delle rappresentazioni di mappe digitali. Consente di descrivere la rappresentazione grafica di dati vettoriali e raster. Solitamente questo standard è utilizzato per istruire un Web Map Services (WMS) su come raffigurare uno specifico layer.

Dall'agosto 2007 le specifiche SLD si sono suddivise in due gruppi di nuove specifiche:

- Symbology Encoding Implementation Specification (SE)
- Styled Layer Descriptor

Attualmente SLD contiene solo il protocollo per comunicare ad un WMS come dovrà raffigurare un layer, mentre la descrizione vera e propria dello stile è esclusivamente delegata ad SE.

Nonostante lo standard SE permetta una descrizione di uno stile molto più completa e con varie funzionalità aggiuntive rispetto alla vecchia versione delle specifiche SLD (ad esempio Displacement, Offset, InitialGap, etc.), non vi sono attualmente molti GIS che lo gestiscono. OpenJump, GIS utilizzato nel progetto, ad esempio, non dispone ancora di un plug-in che permetta di utilizzarlo e questo vale anche per i più comuni GIS sia commerciali che open source.

Un file SLD contiene un elemento XML root chiamato *StyledLayerDescriptor* che racchiude al suo interno una sequenza di elementi detti *styled-layer*.

Un *Layer* è definito come una collezione di feature che possono potenzialmente essere di varie tipologie geometriche. Questi nello standard SLD possono essere *NamedLayer* o *UserLayer*. Data la propensione dello standard verso i servizi WMS (Web Map Service), i primi sono definiti come layer accessibili da un GIS web server usando un "well-known name", mentre l'altra tipologia indica dei layer accessibili grazie a parametri aggiuntivi che permettono di identificare la posizione in rete e le caratteristiche dello stesso permettendo così di essere trovati e visualizzati correttamente.

A ogni layer, di entrambe le tipologie, viene affiancato uno stile. Anche questi possono essere suddivise in due tipologie distinte *NamedStyle* oppure *UserStyle*, che vengono definiti in maniera molto simile ai layer, ovvero i primi sono indicati con un well-known name, e sono rintracciabili attraverso questo all'interno del GIS web server, mentre gli altri sono definiti direttamente con una serie di elementi XML che determinano lo stile che verrà dato al Layer.

Tra i principali software open source che sono compatibili allo standard si possono citare OpenJump e UDig per la sezione Desktop GIS e GeoServer e MapServer per la i Server-Side software.

3 PROGETTO

3.1 INTRODUZIONE

Dopo aver fornito un'introduzione generale alle tematiche affrontate, i paragrafi successivi descriveranno il lavoro da me svolto all'interno di CARGEN.

Durante lo sviluppo degli algoritmi del progetto CARGEN ci si è resi conto della necessità di avere un tool che permettesse di visualizzare i dati creati attraverso qualche funzionalità avanzata, come ad esempio, la possibilità di caricare l'intero database creato, o una particolare porzione dello stesso e potergli assegnare una vestizione grafica che riproducesse fedelmente la simbologia ufficiale.

Nonostante la vestizione del database sembri un argomento che un po' esula dal processo di generalizzazione, esso risulta in realtà una parte integrante e di vitale importanza per una buona realizzazione del lavoro. Come si è visto nel capitolo precedente, infatti, lo scopo dell'intero processo di generalizzazione è quello di modificare i dati iniziali per creare una nuova rappresentazione che permetta di visualizzare correttamente e con la maggiore accuratezza possibile il contenuto informativo della mappa originale ad una nuova scala. In questa ottica, la presenza di uno strumento che permetta di valutare in maniera efficace i dati prodotti dai vari algoritmi di generalizzazione risulta essere di grandissima utilità per poter guidare lo sviluppo dell'intero processo.

3.1.1 OBIETTIVI

In precedenza si è accennato agli obiettivi del mio lavoro e, in questo paragrafo l'argomento verrà approfondito e analizzato più dettagliatamente.

L'obiettivo fondamentale è la creazione di uno strumento che permetta di testare i risultati del processo di generalizzazione rappresentandoli utilizzando una vestizione che rispetti il più fedelmente possibile gli standard rappresentativi del DB25 e DB50, consentendo quindi di individuare problemi ed eventuali migliorie possibili.

Come accennato nell'introduzione, si è reso necessario, inoltre, aggiungere funzionalità che permettessero di caricare all'interno del GIS utilizzato l'intero database o una porzione dello stesso e di applicargli la vestizione, rispettando le regole

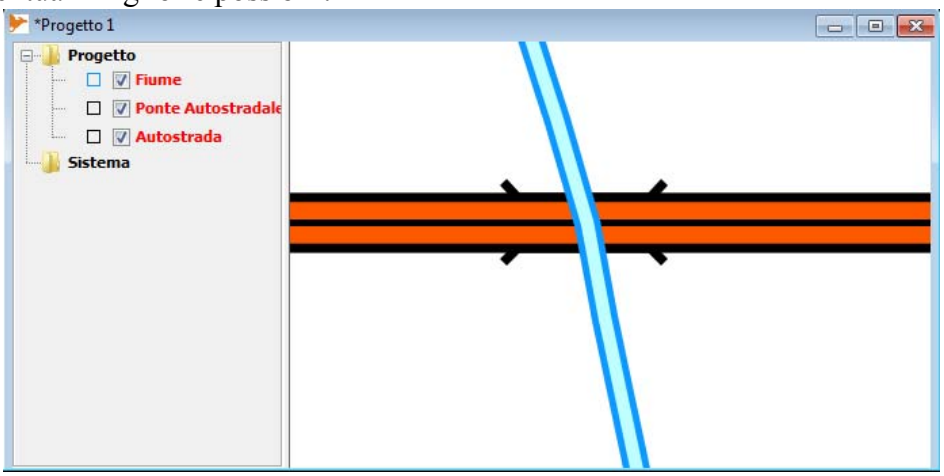

Figura 6: Esempio di cattivà gestione della priorità dei layers

relative alla priorità, senza la quale si otterrebbe una visualizzazione dei dati non corretta.

Dato che gli standard IGM spesso impongono spessori del tratteggio e simbologia di dimensioni molto piccole si è rivelato utile fornire al software prodotto una funzionalità di zoom che quando si modifica la scala, aumenti o diminuisca proporzionalmente la larghezza dei tratteggi, e la grandezza dei simboli. Il tool per lo zoom fornito da OpenJump, infatti, a fronte di un ingrandimento o riduzione della mappa, non modificava lo stile reimpostato.

3.1.2 PERCORSO

Per raggiungere gli obiettivi prefissati si è deciso di creare un plug-in per il desktop GIS OpenJump.

La prima fase del progetto è stata un'attenta analisi dello standard SLD: questo ha fornito le basi per una standardizzazione che permette una completa descrizione degli stili da assegnare alle varie features. Sulla base di ciò è stato impostato un "foglio di stile", un file in formato CSV, che ha il compito di contenere tutte le informazioni relative agli stili da utilizzare nella vestizione. Per strutturare questo foglio sono stati utilizzati come modello NamedLayer e UserStyle. I layer, contenenti ognuno una tipologia diversa di feature e contraddistinti da un codice LAB, ottengono i corrispondenti stili attraverso la compilazione di campi nel foglio, che corrispondono agli elementi descrittivi XML contenuti nei file SLD.

Successivamente questo foglio è stato compilato con tutte le informazioni necessarie per raffigurare la simbologia relativa al DB25 e al DB50.

Dopo la prima parte di analisi e strutturazione, è iniziato il vero e proprio sviluppo del plug-in che consiste principalmente in un parser che permette la traduzione del foglio di stile in due forme di output: un file SLD conforme all'XML Schema dello standard OGC e degli oggetti da utilizzare nella raffigurazione dei dati da visualizzare all'interno di OpenJump.

Lo sviluppo del plug-in è continuato poi con la programmazione di classi e algoritmi che permettono di visualizzare correttamente ogni stile descritto nel foglio di stile, successivamente letto e interpretato dal parser. Inizialmente si è realizzata una funzionalità aggiuntiva per OpenJump che permette di associare a un layer uno stile scelto dall'utente; successivamente è stata aggiunta la possibilità di poter applicare in maniera automatica gli stili IGM alle tabelle Oracle contenenti i dati generalizzati. Durante il processo di lettura delle tabelle è stato inoltre inserito un algoritmo che ordina la visualizzazione delle tabelle in OpenJump in base alla priorità inserita nel foglio di stile per ottenere una buona raffigurazione della mappa digitale richiesta.

3.1.3 TECNOLOGIA UTILIZZATA

OpenJump

Il software ottenuto è un plug-in per **OpenJump,** un desktop GIS open source scritto in linguaggio di programmazione Java e distribuito sotto licenza GNU GPL(General Public License).

Figura 7

Di seguito si elencano le caratteristiche che lo contraddistinguono:

- È indipendente dalla piattaforma;
- Legge e scrive ESRI ShapeFile, file GML, DXF e PostGis;
- Possiede numerosi plug-in sviluppati da terzi, per connettersi a PostGis, Oracle database o AcrSDE, per stampare, etc.;
- Legge file raster quali TIFF, JPEG, PNG e ECW;
- Permette l'editing di attributi e di qualsiasi geometria;
- Possiede algoritmi geometrici basati su Java Topology Suite;
- Supporta standard come WMS, WFS e SLD.

Questo programma è stato scelto per la semplicità, la completezza e soprattutto la versatilità. Una caratteristica fondamentale del software, infatti, è quella di poter sviluppare plug-in da inserire all'interno del GIS molto facilmente, potendo anche utilizzare le molte funzioni di cui già dispone per ottenere il risultato desiderato.

Eclipse

Il plug-in è stato sviluppato grazie alla piattaforma IDE (Integrated Development Environment) **Eclipse**, un progetto open source che ha come scopo la creazione e l'implementazione di una piattaforma di sviluppo. Ideato nel 2001 da un consorzio no-profit di grandi società (IBM, Borland, Red Hat, SUSE, Ericsson, HP, Intel, SAP etc.) chiamato Eclipse Fondation, è supportato da una gremita comunità strutturata sullo stile open source. È interamente scritto in Java e ciò lo rende multipiattaforma, infatti, è disponibile per Windows, Linux, MAC OS X

Figura 8

Le funzionalità che la piattaforma mette a disposizione sono state indispensabili per la creazione del programma, per una buona gestione del'intero progetto, per l'interazione e quindi per l'integrazione del software creato con quello preesistente. Tra le funzioni più utilizzate e maggiormente apprezzate si possono citare il completamento automatico, il suggerimento di parametri e metodi, i facili

collegamenti e suggerimenti derivanti dai JavaDoc collegati alle varie funzioni e infine le funzioni per la creazione automatica di codice e di ristrutturazioni dello stesso.

Java e JTS

Il plug-in è stato sviluppato in **Java**, in quanto OpenJump è basato su questo linguaggio di programmazione. La versione del JDK (Java Development Kit) è la 1.6.0_19.

Figura 9

Java è un linguaggio di programmazione nato negli anni Novanta, inizialmente per scopi di ricerca in ambito universitario. È orientato agli oggetti ed è molto efficace per la creazione di applicazioni sicure. Le principali caratteristiche di Java sono l'indipendenza dalla piattaforma utilizzata grazie alla JVM (Java Virtual Machine) e una sintassi molto simile al linguaggio C, ma completamente object-oriented.

Per sviluppare il software sono state utilizzate, oltre alle librerie standard, anche quelle **JTS (**Java Topology Suite**)**.

JTS è un API di funzioni spaziali 2D open source e sotto licenza LGPL. Queste librerie sviluppate da Vivid Solution sono scritte completamente in Java, sono conformi allo standard Simple Features Specification for SQL pubblicato da Open GIS Consortium e forniscono una completa e consistente implementazione dei fondamentali algoritmi spaziali 2D. Queste hanno permesso la creazione e la manipolazione delle geometrie da vestire, permettendo di raggiungere l'effetto grafico desiderato nella vestizione, altrimenti non ottenibile con le funzioni standard.

3.2 FOGLIO DI STILE

Il foglio di stile è un file CSV che permette di creare un nuovo stile compilando i vari campi che si riferiscono alle varie caratteristiche desiderate. Il file è strutturato in modo molto semplice. Le prime dieci colonne sono necessarie al parser per rendere univoco lo stile e per ottenere le informazioni per quanto concerne la parte del foglio che dovrà analizzare e le eventuali decorazioni aggiuntive che verranno applicate alle varie geometrie. Le altre colonne invece rispecchiano la strutturazione gerarchica dello standard SLD.

Il foglio di stile è scomponibile in 4 parti ben distinte, che verranno approfondite separatamente nei paragrafi successivi, permettendo così una più facile comprensione sia dell'utilizzo del foglio per la creazione di uno stile personalizzato, sia per poter sfruttare tutte le potenzialità che lo strumento possiede.

3.2.1 ATTRIBUTI ASSOCIATI AL CODICE LAB DI UNA FEATURE

5	NAMED LAYER									
6	FACC	Gruppo	LAB	Nome	Priorità	JTS	JTS 2	Duplicazioni	Symbolizer	Appunti
7										
8										
9										
10										
11	AN010		L201	Ferrovia a	9	100,131,10	#00000007	1	linea	linea
12	AN010		L202	Ferrovia a	9	100,131,1(	100,131,1(	1	linea	linea
13	AN010		L205	Ferrovia a	9	110,130,40	#00000007	1	linea	linea
14	AN010		L207	Tranvia in	9			2	linea	linea
15									linea	linea

Figura 10: Screenshot della parte iniziale del foglio di stile contenente gli attributi delle feature

Gli elementi DB25 e DB50, che sulla base dell'analisi svolta per la generalizzazione risultano avere gli stessi attributi e caratteristiche simili, sono raccolti nel database all'interno di tabelle denominate da un univoco codici **FACC**. La compilazione di questo campo è necessaria ai fini dell'ordinamento per priorità.

A ogni elemento è associato anche un codice **LAB** che permette di identificare a quale categoria di elementi topografici appartiene (es. autostrada, strada extraurbana principale, strada extraurbana secondaria). A ogni LAB è associato il **Nome** dell'elemento indicato nel campo successivo.

Il campo **Priorità** viene compilato con una numerazione compresa tra 1 e 14 ed è necessaria per ordinare varie categorie di layer che così vengono visualizzati in maniera corretta, evitando problemi dovuti alla sovrapposizione di simboli che potrebbero andare a nascondere parzialmente o completamente altri vicini o che si trovano all'interno della geometria dei primi. La priorità è data in ordine decrescente, quindi le features con priorità 14 verranno raffigurate sopra tutte le altre.

A seconda della complessità di un simbolo associato al codice LAB può essere necessario utilizzare uno stile che composto da più stili semplici, uno sovrapposto all'altro. Un classico esempio è quello della simbologia per rappresentare un'autostrada, composto da tre stili semplici: uno nero di spessore uguale a quello dell'intera linea, uno arancione, di spessore leggermente più piccolo stampato sopra, dando così l'effetto delle sue bande nere ai bordi della linea, e una linea nera di spessore ancora minore, per visualizzare la linea centrale. Per questo motivo è stato introdotto il campo **Duplicazioni**, che ha il compito di comunicare al parser da quante righe del foglio CSV è composto lo stile. Occorre sottolineare che viene utilizzato, proprio come SLD, il *painter-model*. Questo significa che gli stili che vengono disegnati per primi, possono essere coperti dagli stili che verranno applicati dopo di loro.

Figura 11: Simbologia utilizzata per le autostrade

Il campo **Symbolizer** deve essere completato per ogni riga con una tra le seguenti quattro stringhe: *"punto", "linea", "poligono", "??"*. Grazie alle prime tre scelte il parser potrà essere instradato verso la corretta parte del foglio contente la descrizione dello stile. L'opzione "??" è stata inserita per fare in modo che il parser scarti lo stile in fase di analisi degli stili disponibili e faccia in modo che non sia presente nemmeno all'interno del file SLD.

Le colonne **Appunti** e **Gruppi** non sono necessarie ai fini del funzionamento. La prima è stata inserita per fornire uno spazio per degli eventuali commenti del compilatore. La seconda è stata inserita in previsione di un eventuale sviluppo del progetto, per poter raggruppare più tipologie di feature e quindi diversi codici LAB e FACC, per poter ottenere una legenda più comoda e ordinata o per filtrare diverse categorie di stile.

I campi **JTS** e **JTS2** servono per aggiungere alcune simbologie non rappresentabili attraverso attributi contenuti nello standard SLD o altre ottenibili, ma con eccessivi sprechi di risorse o con imprecisioni. Questi simboli aggiuntivi sono stati sviluppati con algoritmi che sfruttano le librerie Java Topology Suite. Ogni stile e le sue caratteristiche sono indicate con diversi codici che verranno descritti nel paragrafo successivo.

Codici JTS

Ogni codice ha in comune le prime due cifre. La prima indica la prima categoria di stili, la seconda invece la tipologia.

- **Codice 0X: Alette**

Nome	Larghezza stroke simbolo (mm)	Lunghezza Alette (mm)	Colore	Larghezza stroke sottostante (mm)
0X	X,XX	X,XX	#XXXXXX	X,XX

Questa prima categoria ingloba al suo interno un tipo di simbologie necessarie per rappresentare correttamente gallerie, ponti e chiuse secondo le specifiche grafiche imposte dagli standard IGM.

- *Codice 00: Gallerie*
  Il simbolo è formato da una LineString composta da 3 segmenti.

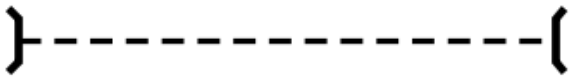

Figura 12: Simbologia usata per le gallerie ferroviarie

  La lunghezza di quelle esterne è dettata dalla misura indicata dal terzo campo del codice e hanno inclinazione di 45°. La LineString centrale unisce quelle esterne e la misura è espressa dal quinto campo. Utilizzando questo codice il disegno verrà applicato sia nel punto iniziale, sia in quello finale della geometria lineare. Il tutto verrà colorato con il colore e con un tratteggio di larghezza deciso dall'utente compilando il secondo e il quarto campo.

- *Codice 01: Ponti*
  La modalità di creazione di questo simbolo è molto simile a quella della precedente, infatti, l'unica cosa non presente è la linea di congiunzione tra le due alette. Per una migliore resa grafica è stato inserito un piccolo offset negativo, che sposta il punto iniziale delle alette verso l'interno e abbassa l'altezza dell'aletta, in entrambi i casi, di un valore corrispondente alla metà della larghezza dello stroke applicato alla stessa.

Figura 13: Simbologia utilizzata per i ponti ferroviari

- *Codice 02: Chiuse*
  È formato da alette esattamente uguali rispetto alla tipologia precedente, e da dei quadrati pieni. La misura del loro lato risulta essere la lunghezza dell'aletta divisa per un fattore √2.

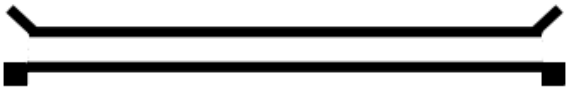

Figura 14: Simbologia per rappresentare le chiuse con passerella

- **Codice 1X: Tratteggi**
  Gli attributi necessari per la creazione del codice sono:

Nome	Larghezza Stroke simbolo (mm)	Lunghezza Tratteggio (mm)	Colore	Distanza fra Tratteggi (mm)	Offset (mm)	Larghezza Stroke sottostante (mm)	Verso	Riemp.
1X	X,XX	X,XX	#XXXXXX	XX,XX	X,XX	X,XX	X	X

- *Codice 10: Trattino completo*
  Viene inserito delle linee di lunghezza impostata nel terzo campo con uno stroke e un colorazione decisa rispettivamente nel secondo e quarta parte del codice sopra proposto. Gli altri campi sono necessari per personalizzare maggiormente lo stile impostando la misura della distanza tra una linea e l'altra (campo 5) e un eventuale offset per far stampare la prima linea in un punto preciso deciso dall'utente. Questo è necessario per poter adattare il tratteggio a eventuali stili che posseggono un *DashArray,* che verrà spiegato nel capitolo successivo.
  Per convenzione, valida per tutti i tipi di tratteggio, se l'offset viene impostato a 0, il primo simbolo verrà posizionato a metà della distanza tra due tratteggi.
  Gli ultimi due campi non sono necessari per questo codice, e possono essere compilati con qualsiasi valore numerico.

Figura 15: Simbologia usata per rappresentare ferrovie in costruzione

- *Codice 11: Trattino orientato*

  Per ottenere un effetto come quello illustrato dalla figura 13 è necessario specificare anche il campo 8 per indicare al parser se la linea deve essere verso l'alto oppure verso l'alto. Per il primo caso bisogna inserire il valore 0, altrimenti il valore 1.

Figura 16: Simbologia utilizzata per rappresentare ferrovie a scartamento ridotto

- *Codice 12: Pallini*

  Il simbolo raffigurato nell'immagine a fianco è un esempio di utilizzo del tratteggio con i pallini. In questo caso l'effetto è composto compilando entrambi i campi JTS, il primo per ottenere i cerchietti sopra, e il secondo per quelli sotto. I campi compilati sono gli stessi del codice 11, con la sola aggiunta dell'ultimo campo presentato nella tabella che è stato settato a 1 per far si che il pallino sia pieno. Per visualizzare un simbolo vuoto bisogna impostare un valore 0.

Figura 17: Simbologia utilizzata per elettrodotti doppi

- *Codice 13: Quadrati*

  L'effetto è stato realizzato attraverso l'utilizzo di quadrati vuoti, applicando un offset pari a 0, in questo caso il valore applicato al campo Verso può essere impostato con un qualsiasi valore dato che è comunque inutile ai fini della rappresentazione.

Figura 18: Simbologia utilizzata per acquedotti in costruzione

- *Codice 14: Triangoli*

  Il simbolismo ottenuto è stato realizzato applicando un offset uguale alla somma di metà stroke della linea e di metà della dimensione del triangolo. Quando si utilizzano figure geometriche per ottenere un determinato tratteggio bisogna considerare che applicando l'offset si sposta il centroide del simbolo.

Figura 19: Simbologia utilizzata per rappresentare cave di tufo

- **Codice 2X: Vertici**

Nome	Grandezza simbolo (mm)	Colore	Offset (mm)	Segno
2X	X,XX	#XXXXXX	X,XX	X

- *Codice 20: Quadrati ai vertici iniziali e finali*

Figura 20:Simbologia utilizzata per rappresentare chiuse senza passerella

Questa tipologia è semplice da realizzare poiché basta inserire la grandezza del simbolo, il colore e un eventuale offset. Con il campo Segno si decide l'orientamento. 0 indica verso un orientamento in senso orario, 1 in senso antiorario. Non è stato inserito un campo per creare un simbolo pieno o uno vuoto, poiché nessun tipo di simbolo osservato nei documenti ufficiali IGM lo richiedeva.

- *Codice 21: Pallini ogni vertice*

Figura 21: Simbologia utilizzata per rappresentare elettrodotti

Come si può vedere dall'immagine 18, il codice rende possibile aggiungere a ogni vertice di un segmento che forma una LineString un cerchietto pieno di dimensione e colore impostati nel codice. In questo caso l'offset è impostato a 0.

3.2.2 POINT SYMBOLIZER

POINT SYMBOLIZER									
External Graphic		Mark						Rotation	Size
OnlineResource	Format	WellKnownName	Fill	Stroke					
			Color	Colore	width	dasharray	dashoffset		
		square		#000000	0,13				0,6
		circle	#000000	#000000	0,01				0,33
1Campo	image/svg+xml								2,3

Figura 22: Screenshot campi relativi al point symbolizer nel foglio di stile

La sezione Point Symbolizer permette la rappresentazione dei simboli puntuali. Sono raffigurabili due differenti tipologie di simboli a seconda che venga compilata la parte **External Graphic** oppure **Mark**.

Se si seleziona la prima opzione è possibile visualizzare un'immagine esterna ad OpenJump scelta dall'utente. Per far ciò è necessario indicare nella relativa cella il nome del file da caricare. I file gestiti devono essere in formato SVG (Scalable Vector Graphics). La seconda colonna **Format**, anche se non è strettamente necessario per il funzionamento del Parser, è stata inserita per una maggiore completezza e per descrivere l'omonimo elemento SLD, necessario se il file viene esportato, e per lasciare spazio a un'eventuale espansione di compatibilità verso altri formati al momento non gestiti dal plug-in. Per creare un file SVG utilizzabile all'interno del plug-in è necessario che questo sia un quadrato e che vengano mantenute le proporzioni rispetto all'immagine che si vuole rappresentare. Sarà poi il programma che adatterà le dimensioni della raffigurazione a seconda della scala.

Nel secondo caso si sceglie un simbolo **Mark**, ovvero un simbolo predefinito che verrà creato grazie a geometrie utilizzando JTS. La cella **WellKnownName** dovrà contenere una tra le seguenti parole chiave: "*circle*", "*square*", "*star*", "*triangle*", "*cross*". A seconda della tipologia scelta verrà visualizzato rispettivamente un cerchio, un quadrato, una stella, un triangolo oppure una croce. Queste potranno essere ulteriormente personalizzate utilizzando un **Fill** per riempirle con il colore desiderato che dovrà essere inserito in formato RGB (Red Green Blue) e uno **Stroke**, ovvero un tratteggio delle linee esterne descritto da quattro campi:

- *Color:* per colorare il tratteggio con il colore desiderato;
- *Width*: per impostare lo spessore della linea;
- *DashArray:* che permette di intervallare spazi regolari colorati e bianchi;
- *DashOffset:* necessario se si vuole impostare il punto di partenza del tratteggio.

Per entrambe le opzioni bisogna impostare il campo **Size** per decidere la dimensione del simbolo che verrà visualizzato. Non vi è distinzione tra larghezza e altezza dell'immagine perché viene comunque adattata ad un quadrato di lato "size".

Se necessario vi è anche la possibilità di orientare il simbolo attraverso l'inserimento dell'angolo desiderato nel campo **Rotation.** Il valore deve essere espresso in gradi.

3.2.3 LINE SYMBOLIZER

LINE SYMBOLIZER													
Stroke				Graphictroke									
Color	width	dasharray	dashoffset	External Graphic		Mark						Rotation	Size
				OnlineResource	Format	WellKnownName	Fill	Stroke					
							Color	Color	width	dasharray	dashoffset		
#000000	0,13												
				1Croce	svg								4,64
#000000	0,61	0.18 0.35											

Figura 23: Screenshot della parte del foglio di stile relativa al line symbolizer

Come si può intuire dal nome, questa sezione del foglio di stile viene utilizzata per impostare uno stile per una linea. Rispetto al Point Symbolizer non compaiono campi nuovi, ma è diverso il loro raggruppamento. Se si vuole ottenere uno stile "semplice", ovvero come quello visto nel paragrafo precedente per i Mark, basta semplicemente compilare i valori relativi a **Stroke.**

In parallelo allo standard SLD, viene data la possibilità di utilizzare un **Graphic Stroke**, ovvero un tratteggio formato da vari simboli puntuali che vengono ripetuti uno dietro l'altro con una frequenza dettata dalla grandezza del campo Size. Per ottenere questo effetto grafico è necessario compilare i campi sotto Graphic Stroke come precedentemente descritto. È sconsigliato comunque abusare di questi stili utilizzando immagini ripetute poiché ciò comporta uno spreco di risorse che può rallentare il sistema. Proprio per questo motivo si è imposta la necessità di utilizzare i campi JTS per ottenere varie semplici decorazioni, che risultano computazionalmente più leggere, e che permettono di ottenere risultati migliori, se l'immagine non è perfettamente centrata.

3.2.4 POLYGON SYMBOLIZER

POLYGON SYMBOLIZER										
Fill	GraphicFill									
Color	External Graphic		Mark						Rotation	Size
	OnlineResource	Format	WellKnownName	Fill	Stroke					
				Color	Color	width	dasharray	dashoffset		
#BFFFFF										

Figura 24: Screenshot della parte del foglio di stile relativa al polygon symbolizer

La sezione Polygon Symbolizer è divisa in due parti, una formata da **Fill** e **Graphic Fill**, e l'altra composta da **Stroke** e **Graphic Stroke**. L'utilizzo di quest'ultima è perfettamente identico al Line Symbolizer, lo stile descritto, infatti, viene applicato alla linee che compongono il margine del poligono.

La prima parte, invece, permette di personalizzare l'area interna della geometria sia applicando un colore nella sezione Color di Fill, selezionando un'immagine o un simbolo Mark che verrà impostato come pattern e ripetuto all'interno del perimetro del poligono. La compilazione del

Graphic Fill avviene esattamente come per Graphic e Graphic Stroke descritti nei paragrafi precedenti.

3.3 PARSER

Il **parser** ha il compito di interpretare le informazioni relative agli stili e a trasformarle in oggetti Java appartenenti alla classe *Feature*. È formato da due classi *CsvReader* e *CsvParser*. La prima contiene le funzioni che permettono di leggere all'interno del file CSV, mentre la seconda utilizza queste funzioni per ottenere una lista di oggetti Feature e produce grazie a questa un file **style.sld**, contenente un XML che rispetta lo XML Schema SLD, utilizzabile per assegnare gli stili anche con GIS, diversi da OpenJump, che supportano il formato

Per creare il file viene utilizzata la funzione toSld() della classe Feature, che richiama al suo interno un altro metodo della stessa natura per tutti gli elementi *Rule* che lo compongono. All'interno di Rule sono contenuti il tipo di Symbolizer utilizzato e un array di stringhe racchiude al suo interno tutti gli attributi di ogni stile, di lunghezza corrispondente al tipo. Per rendere più leggibile il file prodotto è stata creata una classe *OutStamp*, incaricata di reindirizzare l'output al file sld, e che aggiunge una indentazione che può essere nel caso modificata a seconda dell'effetto che si vuole ottenere. Lo stile di default utilizza un TAB, ogni volta che viene creato un elemento ("figlio") che risulta essere attributo di un altro elemento ("padre").

Un'altra classe indispensabile è *MmToPixel*. Si è resa necessaria per tradurre i valori espressi in millimetri, presenti nel foglio di stile, in altri che raffigurassero la stessa figura nei pixel del monitor. Questo è stato realizzato attraverso un algoritmo che legge la risoluzione del monitor utilizzato espressa in dpi (dot per inches) e calcola quanti pixel sono necessari per ottenere una rappresentazione il più simile possibile a quella su carta. È da sottolineare come il risultato spesso non corrisponda all'effettiva rappresentazione ottenibile su formato cartaceo, poiché i pixel di un monitor solitamente a 72 o 96 dpi, non possono raggiungere la precisione di un supporto cartaceo stampato a 300 dpi. Appare evidente, infatti, che una mappa su carta è molto più precisa, rispetto a una visualizzata su schermo, soprattutto per scale piccole come 1:50000, dove molte linee di norma dovrebbero essere tracciate a 0,13 mm, misura che in un monitor non è rappresentabile correttamente dato che un pixel è grande circa 0,32 mm.

3.4 COMPLEX STYLE

Il file SLD creato dal Parser soddisfa appieno le specifiche dettate dallo standard Styled Layer Descriptor e permette di sfruttare tutte le potenzialità offerte dallo standard. Alcuni GIS, come ad esempio OpenJump stesso, non riescono, però, a visualizzare completamente e correttamente tutti gli attributi assegnati allo stile in esso contenuto, a causa di una implementazione non completa dello standard.

Proprio per questo motivo è stato necessario creare una classe *ComplexStyle* che permettesse di gestire tutti gli attributi e le duplicazioni di stile, e che estendesse le classi preesistenti, permettendo così di eliminare le limitazioni che il programma impone.

Per far sì che gli oggetti ComplexStyle, si adattassero alle funzioni per la creazione di stile già esistenti, è stato necessario estendere la classe preesistente BasicStyle, che non permette però di impostare duplicazioni di stile. Quindi per una maggiore compatibilità nel costruttore della classe si è trasformato il primo stile in un BasicStyle, così che le caratteristiche vengano riconosciute all'interno dello strumento di edizione presente già in OpenJump.

Per creare un oggetto ComplexStyle vengono passati al costruttore un array di Rule con un numero di oggetti pari al numero di duplicazioni utilizzate, due stringhe per passare se presenti i codici JTS e JTS2 , e un valore Integer *initScale*, che permette di impostare la scala (1:25000 o 1:50000) di riferimento dei valori inseriti nel foglio di stile. Per ogni Rule presente nell'array, in relazione ai valori contenuti all'interno di essi, vengono inizializzati uno o più tra i seguenti oggetti: *Fill, ComplexStroke* o *ComplexVertexStyle*, che se presenti verranno inseriti nella esatta posizione dell'array in cui sono contenuti; nel caso contrario verranno inseriti valori null.

La prima classe elencata, è stata sviluppata per contenere sia il colore di riempimento se presente, sia un eventuale *GraphicFill,* i cui campi sono stati descritti precedentemente nella sezione *Polygon Symbolizer*. Per contenere tutte le informazioni a esso relativo si è reso necessario creare un'altra classe *Graphic*, utilizzata, all'occorrenza, anche dagli altri due symbolizer.

ComplexStroke, viene inizializzata per tutti i symbolizer, siano questi Point, Line o Polygon. Nel primo caso viene completato solo la sezione Graphic, mentre per gli altri due è inizializzato solo se necessario. L'oggetto contiene al suo interno un valore booleano così da poter distinguere un punto rispetto alle altre geometrie. Contiene, inoltre, un oggetto per il colore appartenente alla classe standard *java.awt.Color*, e uno di tipo BasicStroke che si rifà alla classe *java.awt.BasicStroke* .

Un ulteriore array è destinato a oggetti *ComplexVertexStyle*. Questi sono delle estensioni della classe astratta *VertexStyle* presente in OpenJump, e destinata a contenere tutte le tipologie di figure che possono essere inserite attraverso le specifiche SLD. Per creare le varie simbologie sono state realizzate le seguenti classi che estendono ComplexVertexStyle:

- *ComplexBitmapVertexStyle*, per gestire immagini provenienti da OnlineResource;
- *ComplexCircleVertexStyle,* per i cerchi;
- *ComplexCrossVertexStyle* per le croci;
- *ComplexSquareVertexStyle* per i quadrati;
- *ComplexStarVertexStyle* per le stelle;
- *ComplexTriangleVertexStyle* per i triangoli.

Tutte queste classi sono state implementate, con l'aiuto del codice già presente nel package *de.latlon.deejump.plugin.style*. Rispetto alla classe VertexStyle, sono state inserite alcune nuove funzioni e variabili che hanno permesso di visualizzare in maniera più precisa le varie figure, di poter gestire al meglio lo zoom, e anche di ruotare un simbolo in base un angolo deciso dall'utente.

È stata predisposta inoltre una classe *ComplexVertexStylesFactory*, che permette di creare un oggetto tra quelli elencati a partire dal WellKnownName passato come parametro.

Ritornando alla descrizione di *ComplexStyle*, è presente anche un array per oggetti *OtherSymbol*, necessaria per tradurre i codici, eventualmente contenuti nei campi JTS nel foglio di stile, in parametri poi utilizzati per ottenere la simbologia desiderata.

La funzione fondamentale della classe ComplexStyle è `paint(Feature f, Graphics2D g, Viewport viewport)` . Questa, presa dall'interfaccia *Style* di OpenJump, ha la funzione di visualizzare sullo schermo tutte le informazioni fino ad ora descritte. Questo metodo viene richiamato ogni qualvolta che viene aggiunto uno stile alle feature oppure viene modificata la scala di zoom nel plug-in. La prima parte della funzione prevede un ciclo che inizialmente apprende il symbolizer e in base a quello selezionato, traduce su schermo le informazioni ad esso relative. La seconda parte prevede la raffigurazione delle simbologie descritte dai codici JTS.

Ogni symbolizer è studiato per poter visualizzare la simbologia per ogni tipo di geometria che lo richiama. Il point symbolizer, se chiamato da un punto visualizzerà il simbolo o l'immagine considerando come centro per la stampa a video il punto stesso, nel caso sia una linea, invece verrà utilizzato il punto che sta a metà della linea. Se invocato da un poligono allora verrà calcolato il centroide della figura e lì verrà stampato il simbolo.

Il metodo appena descritto si avvale dell'ausilio della classe *ComplexStyleUtil*, che contiene le funzioni sia per la visualizzazione, sia algoritmi utili per rintracciare i punti all'interno o adiacenti alle geometrie, su cui basare raffigurazione dei simboli descritti dai codici.

Una caratteristica fondamentale di Complex Style è la gestione dei cambiamenti di scala. Ogni qualvolta che viene questa viene modificata, infatti, lo stile si adatta ad essa. Ad esempio, se si aumenta la scala, la larghezza dello stroke aumenterà proporzionalmente alla scala visualizzando un tratteggio più spesso, rispetto a quello visualizzato alla scala iniziale.

3.5 PLUGIN

Dopo la panoramica sul funzionamento interno di Parser e ComplexStyle, è necessario descrivere il funzionamento del plug-in che sfrutta il lavoro degli elementi appena descritti per soddisfare gli obiettivi posti all'inizio del progetto.

Il plug-in propone due funzionalità:

- **Single Layer**
- **DB Layer**

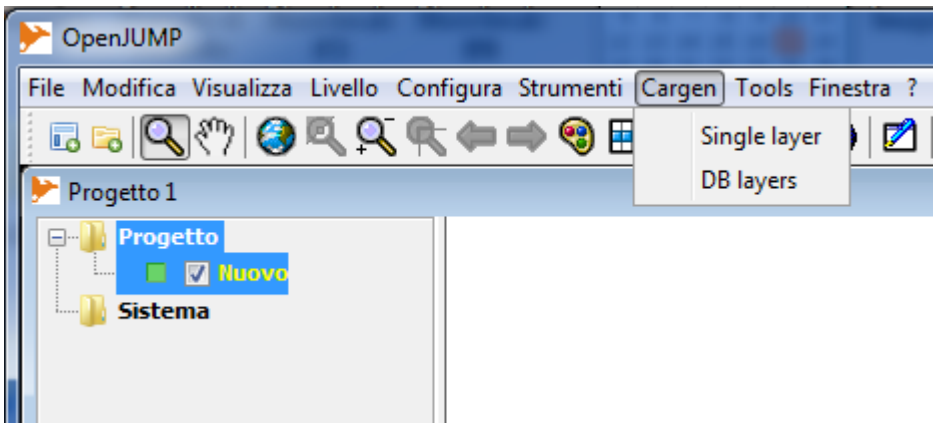

Figura 25: Screenshot della barra dei menù contenente l'icona del plug-in sviluppato

3.5.1 SINGLE LAYER

Single Layer permette di scegliere uno stile da un foglio di stile e di vestire con esso un qualsiasi layer già presente nel workbench di OpenJump.

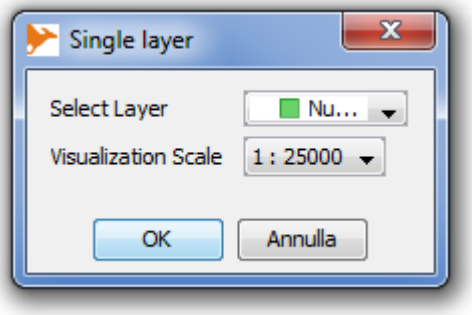

Figura 26: Screenshot della schermata iniziale della funzione Single Layer

Per accedere alla funzionalità è necessario che almeno un layer sia presente all'interno di OpenJump. Dopo aver selezionato il giusto campo sulla barra dei menù, compare per prima cosa una finestra con due menù a tendina (Combo Box), che danno la possibilità di decidere a quale layer assegnare lo stile e la scala cui i valori contenuti all'interno del foglio di stile fanno riferimento. Non appena viene premuto il bottone OK, appare una finestra FileChooser che consente di selezionare il path dove reperire il foglio di stile da passare al parser.

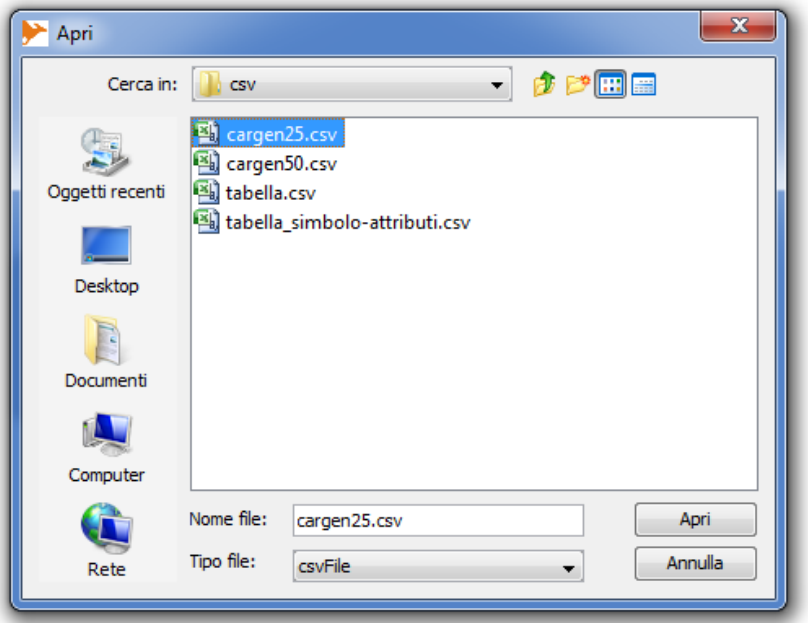

Figura 27: Screenshot della schermata per selezionare il file CSV

Dopo aver selezionato il file e aver premuto il tasto apri, il plug-in passa l'ubicazione del file scelto al parser che ha il compito di creare per ogni stile una lista di feature contenente le informazioni relative allo stesso.

Fatto ciò si presenta in un'altra finestra l'intera lista di stili presenti nel foglio di stile in ordine di apparizione all'interno del file.

Dopo aver selezionato lo stile che si desidera applicare e aver confermato la scelta premendo il pulsante Set, il plug-in provvederà a creare un nuovo ComplexStyle sulla base delle informazioni contenute all'interno dell'oggetto feature relativo al nome selezionato, e applica così lo stile al layer precedentemente scelto.

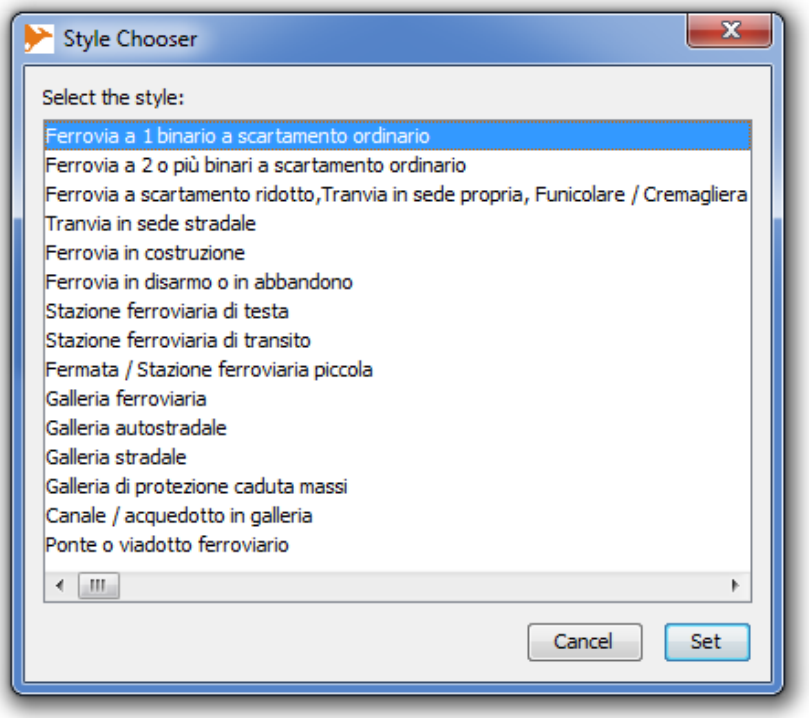

Figura 28: Screenshot dello Style Chooser che permette di selezionare lo stile desiderato tra quelli creati nel file CSV

3.5.2 DB LAYERS

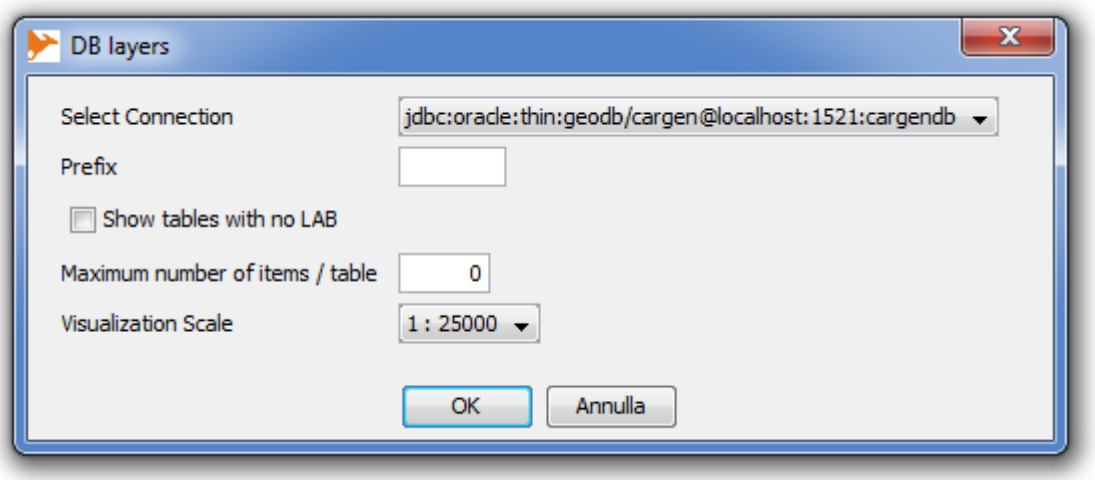

Figura 29: Screenshot della schermata iniziale della funzione DB Layers

Mentre Single Layer si applica a un layer qualunque, DB Layers, come indica il nome stesso, si occupa non solo di applicare degli stili, ma anche di reperire le informazioni da un database, per creare le feature e quindi i layer che le contengono. La finestra principale, che si apre selezionando la funzione dalla barra dei menù si compone (come illustrato nella figura 26) da una Combo Box per selezionare la connessione al database da utilizzare per reperire le tabelle contenti i dati da

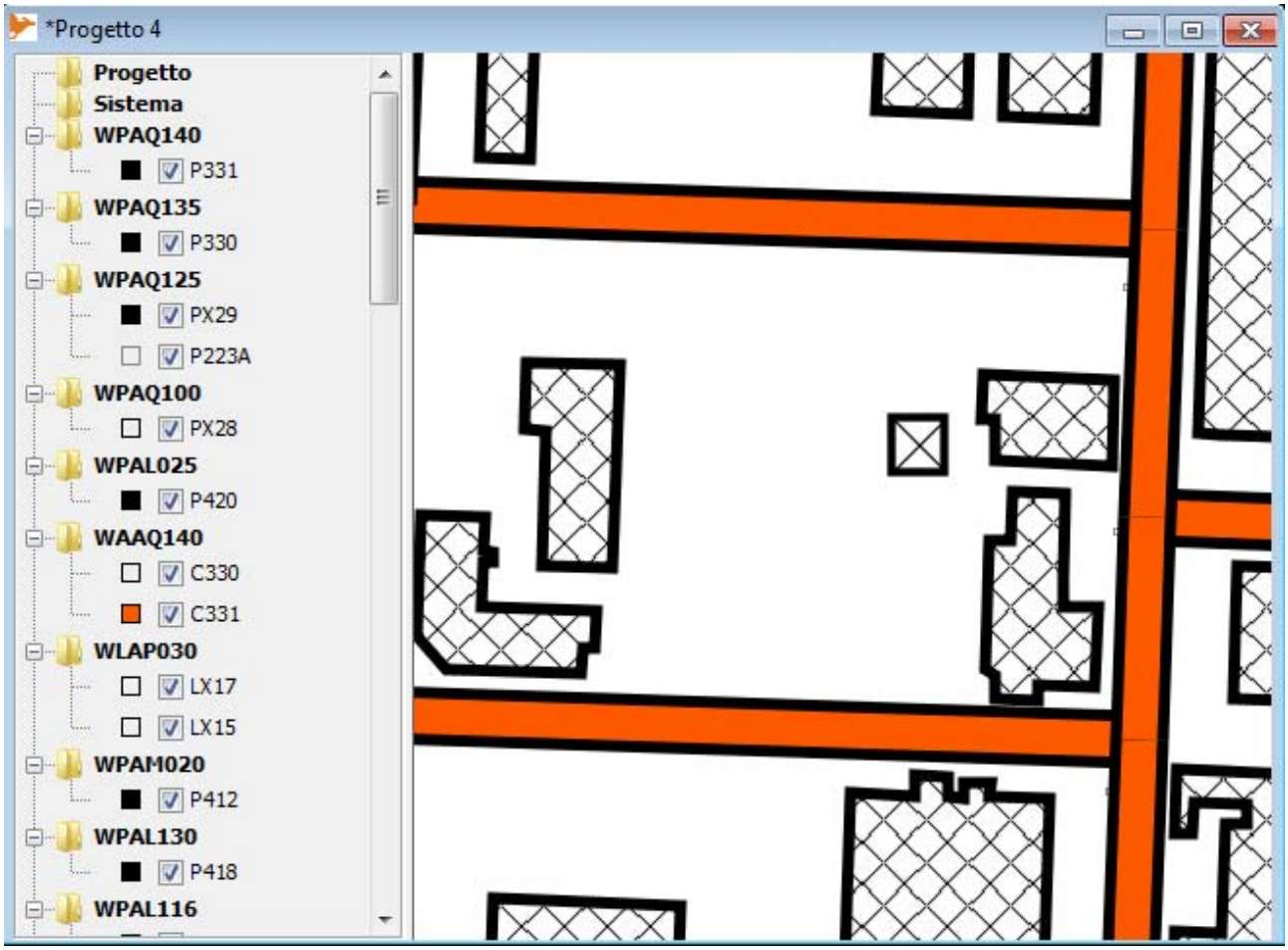

Figura 30: Porzione del DB25 caricata e vestita con DB layers

visualizzare. Tutte queste tabelle sono denominate con un codice contenente un prefisso e un codice FACC. Attraverso l'uso del campo *Prefix* è possibile limitare il numero di tabelle che verranno caricate in OpenJump a quelle il cui nome ha un determinato prefisso; è inoltre possibile selezionare anche solo una singola tabella inserendo nel campo il suo nome completo. Data la varietà di features e di tabelle presenti nei vari database del laboratorio GIRST, è possibile che alcune tabelle contengano feature privi di codice LAB. In questo caso non ci sarà uno stile corrispondente. Con il Check Box aggiunto come terzo campo è possibile decidere se caricare in memoria anche quelle feature con uno stile di default, oppure se non considerarle. Per la vastità di dati raccolti nelle table dei database, si è resa utile una limitazione aggiuntiva, che permette di decidere un numero massimo di feature da inserire in un layer relativo a un singolo codice LAB. Proprio come su Single Layer è necessario impostare la scala e, dato l'OK, bisognerà selezionare il path del file CSV da utilizzare, così inizia il caricamento delle features.

Per effettuare la connessione è stata utilizzata la classe *DBUtility* che ha permesso semplificare l'operazione attraverso funzione già reimpostate.

La prima operazione svolta dal plug-in è una query al database che restituisce tutti i nomi delle tabelle, su cui viene fatta una selezione in base al prefisso, e un successivo ordinamento in base alla priorità che viene appresa dalla cella del foglio di stile, corrispondente a un singolo codice LAB. Dato che le tipologie inserite in una tabella FACC hanno tutte la medesima priorità è necessaria una singola lettura per ogni tabella.

A ogni FACC è associata una Categoria, ovvero un oggetto all'interno di OpenJump, che viene visualizzato come una cartella nella legenda che contiene tutti i layer. Ogni layer conterrà tutte le feature contraddistinte da un singolo LAB, a cui verrà attribuito il corrispondente stile.

L'ordinamento avviene in ordine decrescente, ovvero le tabelle aventi priorità più alta verranno visualizzati al di sopra di quelli con priorità minore. Questo si è reso necessario a fronte di un problema di visualizzazione che se non gestito avrebbe comportato la cancellazione totale o parziale di alcuni simboli, con relativa perdita di contenuto informativo.

4 CONCLUSIONI

4.1 RISULTATI

Il progetto ha conseguito i seguenti risultati:

- Fornisce un metodo per la creazione di stili veloce e semplice da utilizzare.
- Sono stati creati i fogli di stile per la vestizione rispettivamente di DB25 e DB50, utilizzando la simbologia dettata nelle specifiche IGM, coprendo la vestizione di oltre 200 tipologie diverse di feature.
- E' stato realizzato un parser che permette di codificare i fogli di stile secondo lo standard SLD; in particolare sono stati prodotti due file SLD contenente la vestizione IGM per il DB25 e DB50, rendendo queste esportabili ad altri GIS diversi da OpenJump.
- È stata implementata una classe che permette di superare le limitazioni sulla creazione di stili dettate da OpenJump e dallo standard SLD stesso.
- È stato modificato il metodo di visualizzazione dei dati di OpenJump, rendendo possibile la visualizzazione della simbologia sia in paper unit che in ground unit. In modalità paper unit le dimensioni dei simboli sono riferiti ai pixel, pertanto al variare del livello di zoom della mappa essi non cambiano. In modalità ground unit le dimensioni dei simboli sono riferite alla scala della mappa e quindi, al variare del livello di zoom adottato, essi aumentano o diminuiscono di dimensione mantenendo costante la loro misura proiettata al suolo.

Figura 31: Astevia ed edifici generici di una porzione della mappa di Monselice vestita con il plug-in

- È stata fornita una funzione che permette di vestire qualsiasi layer già caricato in OpenJump, con uno stile qualsiasi tra quelli disponibili.

- È stato creato un plug-in che permette di selezionare un intero database o una sua parte, e di vestirlo in maniera automatica.

4.2 PRESTAZIONI

Il plug-in soddisfa effettivamente gli obiettivi preposti.

Testando il risultato appare evidente come i vantaggi ottenuti siano notevoli rispetto alla situazione iniziale. Il programma non appare appesantito dalle nuove funzionalità: i tempi di caricamento degli stili sono leggermente superiori, ma questo è dovuto al fatto che sono più complessi rispetto a quelli standard, essendo composti da più duplicazioni che comportano la creazione di due o più stili standard.

Molte delle limitazioni raffigurative sono state superate con algoritmi per la creazione di nuova simbologia, quali la classe OtherSymbol che permette di tradurre i codici JTS e ComplexStyle che li visualizza a schermo.

È da sottolineare che i tempi di attesa del caricamento dei database possono superare anche la manciata di minuti, ma questo è dovuto alla ingente mole di informazioni che vengono trasmesse dal database contenuto nel server, e trasferiti tramite rete al programma. Non appena tutto è stato caricato, grazie anche alle funzioni di *clip* già presenti in OpenJump, la visualizzazione, lo scorrimento della mappa e lo zoom appaiono comunque operazioni non pesanti, e che richiedono al massimo qualche secondo per visualizzare correttamente le informazioni.

Dal punto di vista prestazionale, l'aspetto più oneroso riguarda l'utilizzo di file grafici per la rappresentazione della simbologia: a causa della gestione delle immagini da parte di OpenJump e ogni volta che viene invocata la funzione *paint()* da parte del programma, tutte le immagini già caricate vengono ricaricate, e questo comporta un aggravio di utilizzo di risorse. In parte l'utilizzo di file grafici SVG è stato sostituito disegnando esplicitamente le geometrie delle vestizioni tramite JTS. Si consiglia quindi di limitare l'uso di immagini esterne, ai casi strettamente necessari, e di trovare metodi alternativi se possibile.

4.3 VANTAGGI

Il plug-in creato, già dai primi test, si è dimostrato un ottimo strumento di supporto per il lavoro svolto nel progetto CARGEN.

Il programma risulta essere un valido aiuto per il "debug" del lavoro di generalizzazione svolto dagli algoritmi: visualizzare i dati con le giuste vestizioni, infatti, permette di trovare più facilmente eventuali errori, e di valutare la qualità dei risultati, aiutando ad identificare dettagli da modificare per ottenere risultati migliori dal processo di generalizzazione.

Un altro vantaggio è dovuto alla possibilità di caricare e vestire un intero database con una singola operazione, cosa che prima necessitava di un esagerato numero di query, che faceva perdere molto tempo e che comunque dava un risultato assolutamente insoddisfacente. Basti pensare che prima dell'introduzione del plug-in, per ottenere una buona visualizzazione era necessario, dopo aver caricato i dati, ordinare i vari layer manualmente, in modo che le informazioni non si sovrapponessero in maniera scorretta, alterando e in alcuni casi eliminando il contenuto informativo della mappa, e successivamente bisognava impostare per ogni layer lo stile esatto. Applicare lo stile a una tipologia di feature, però, comportava diverse operazioni di duplicazione del layer oltre alla definizione dello stile. Moltiplicando questo lavoro per le oltre 200 diverse simbologie presenti al momento attuale nel database, rende evidente il risparmio di tempo e risorse che questo plug-in apporta a coloro che operano all'interno del processo di generalizzazione del progetto CARGEN.

Un altro elemento di fondamentale importanza per la verifica dei dati prodotti, riguarda l'utile strumento di zoom che sostituisce quello originario di OpenJump e permette di mantenere le dimensioni dell'elemento da valutare, anche a fronte di un aumento o una diminuzione di scala. Quando si riduceva la scala, infatti, gli spessori delle linee oppure i simboli mantenevano le loro dimensioni; ciò comportava una sovrapposizione di tutti gli elementi, causando una perdita consistente e spesso totale del contenuto informativo della mappa. Viceversa, se la scala si ingrandiva, comunque le misure dei simboli rimanevano inalterate, comportando una visualizzazione, da un lato effettivamente più chiara, poiché le figure spesso non risultavano più sovrapposte, ma completamente inutile a fini informativi e per valutare la qualità della generalizzazione effettuata.

4.4 SVILUPPI FUTURI

Durante la creazione del progetto sono emerse diverse possibilità di sviluppo per il plug-in.

Prima tra tutte, l'implementazione e la creazione di file che soddisfino il nuovo standard SE, precedentemente descritto. La scelta di non adottare l'ultimo standard proposto da OGC nel progetto attuale è stata presa sulla base di una valutazione sui principali GIS open source e commerciali che non lo supportano, nonostante sia stato standardizzato circa tre anni fa. L'introduzione di questo standard, a nostro parere, potrebbe, però, riuscire a eliminare alcune limitazioni che il vecchio standard impone sulla creazione degli stili, e che ha costretto alla creazione di alcune metodologie, come i campi JTS, per poter ottenere l'intera simbologia IGM, che però non è esportabile al di fuori da OpenJump.

Uno sviluppo più ampio e strettamente legato all'utilizzo di SE è la creazione di un altro plug-in per OpenJump che offra una serie di strumenti utili alla creazione di stili multipli, che rimpiazzi i limitati strumenti per l'edizione di stili presenti all'interno di OpenJump.

Un ulteriore miglioramento, che richiede un dispendio di tempo sicuramente inferiore, ma apporta un buon incremento prestazionale, è la possibilità di sviluppare nuovi codici JTS per evitare la creazione di Graphic Stroke attraverso l'utilizzo di immagini SVG, che effettivamente rallentano la visualizzazione e spesso danno un risultato di scarsa qualità.

In merito all'utilissima funzionalità di zoom, per ottenere una funzione più prestante, sarebbe utile aggiungere un limite di scala, superato il quale il ComplexStyle viene sostituito da un BasicStyle. L'operazione renderebbe più veloce il caricamento della vestizione con le scale più piccole dove un tratteggio semplice, spesso, è talmente piccolo da essere indistinguibile da un tratteggio complesso formato da più duplicazioni.

5 ELENCO FIGURE

Figura 1: Tavoletta con pianta di distretto agricolo, da Nuzi, Mesopotamia....2
Figura 2: Esempio di mappa generalizzata a varie scale 6
Figura 3: Parte della "Tabula Peutingeriana"....13
Figura 5: Porzione di carta geografica vestita....14
Figura 4: Porzione di carta geografica non vestita....14
Figura 6: Esempio di cattivà gestione della priorità dei layers....17
Figura 7: Logo OpenJump 19
Figura 8: Logo Eclipse....19
Figura 9: Logo Java e Vivid Solution 20
Figura 10: Screenshot della parte iniziale del foglio di stile contenente gli attributi delle feature....21
Figura 11: Simbologia utilizzata per le autostrade....21
Figura 12: Simbologia usata per le gallerie ferroviarie....22
Figura 13: Simbologia utilizzata per i ponti ferroviari 23
Figura 14: Simbologia per rappresentare le chiuse con passerella 23
Figura 15: Simbologia usata per rappresentare ferrovie in costruzione....23
Figura 16: Simbologia utilizzata per rappresentare ferrovie a scartamento ridotto....24
Figura 17: Simbologia utilizzata per elettrodotti doppi 24
Figura 18: Simbologia utilizzata per acquedotti in costruzione....24
Figura 19: Simbologia utilizzata per rappresentare cave di tufo 24
Figura 22: Screenshot campi relativi al point symbolizer nel foglio di stile 25
Figura 20:Simbologia utilizzata per rappresentare chiuse senza passerella 25
Figura 21: Simbologia utilizzata per rappresentare elettrodotti....25
Figura 23: Screenshot della parte del foglio di stile relativa al line symbolizer....27
Figura 24: Screenshot della parte del foglio di stile relativa al polygon symbolizer....27
Figura 25: Screenshot della barra dei menù contenente l'icona del plug-in sviluppato 30
Figura 26: Screenshot della schermata iniziale della funzione Single Layer 31
Figura 27: Screenshot della schermata per selezionare il file CSV 31
Figura 28: Screenshot dello Style Chooser....32
Figura 29: Screenshot della schermata iniziale della funzione DB Layers....32
Figura 30: Porzione del DB25 caricata e vestita con DB layers....33
Figura 31: Porzione della mappa di Monselice vestita con il plug-in....35