Webb.IT 03
Padova 09/05/2003

GISToolkit Tutorial
java, gis e software libero

Maurizio Napolitano
napo@itc.it
http://sra.itc.it/people/napolitano

ITC irst

Introduzione

GISToolkit

GIS
Geographical Information System

Toolkit
insieme di strumenti

Un insieme di strumenti (= classi) per lo sviluppo di applicazioni GIS in **JAVA**

Distribuito con licenza LGPL
http://www.gnu.org/copyleft/lesser.html

Lesser General Public License
permette la creazione di software non libero a patto che la componente LGPL usata venga distribuita con i sorgenti

Breve storia di GISToolkit

Developer team:

BitterStorm: fondatore del progetto nel maggio 2001
Ithaqua: attuale maintainer

BitterStorm: Ha dato vita al progetto creando il "core" principale del toolkit (con particolare attenzione verso la gestione del formato ESRI shp).
Ha abbandonato il progetto in favore di un altro.
Contribuisce ancora allo sviluppo

Ithaqua: Entrato nel progetto mosso dalla necessità di sviluppare una applicazione GIS commerciale in Java.
Non trovando alcuna soluzione commerciale che lo soddisfacesse ha preso in mano GISToolKit contribuendone in maniera concreta allo sviluppo fino a diventarne l'attuale capo-progetto.

GISEditor e GISServer

GISToolkit offre due applicazioni di esempio (una desktop ed una client/server) in grado di mostrare il potenziale di queste classi

GISEditor

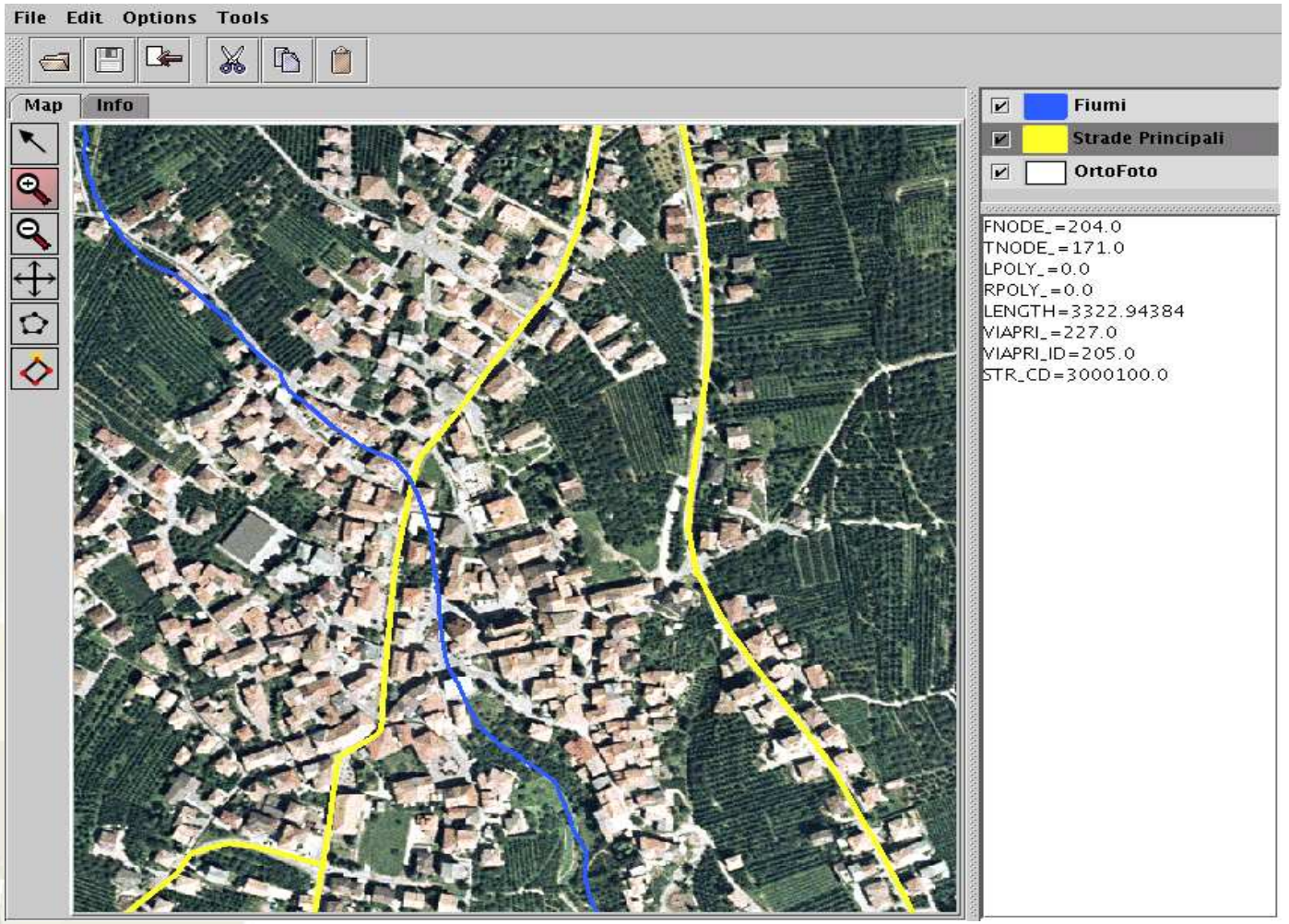
Applicazione desktop in grado di accedere a diverse sorgenti GIS (sia vettoriali che raster), visualizzare i dati, definire gli stili delle viste, cambiare il tipo di proiezione, modificare i dati ...

GISServer

Si tratta di un Web Map Server compatibile con lo standard definito dall'Open Gis Consortium.
Allo stato attuale e' una applicazione molto semplice che permette la visualizzazione di dati vettoriali (via HTML o via Applet Java) di cui sono stati definiti i layout nella parte amministrativa

GISEditor

```
java -Xms32m -Xmx128m gistoolkit.application.GISEditor
```



GISServer (admin)

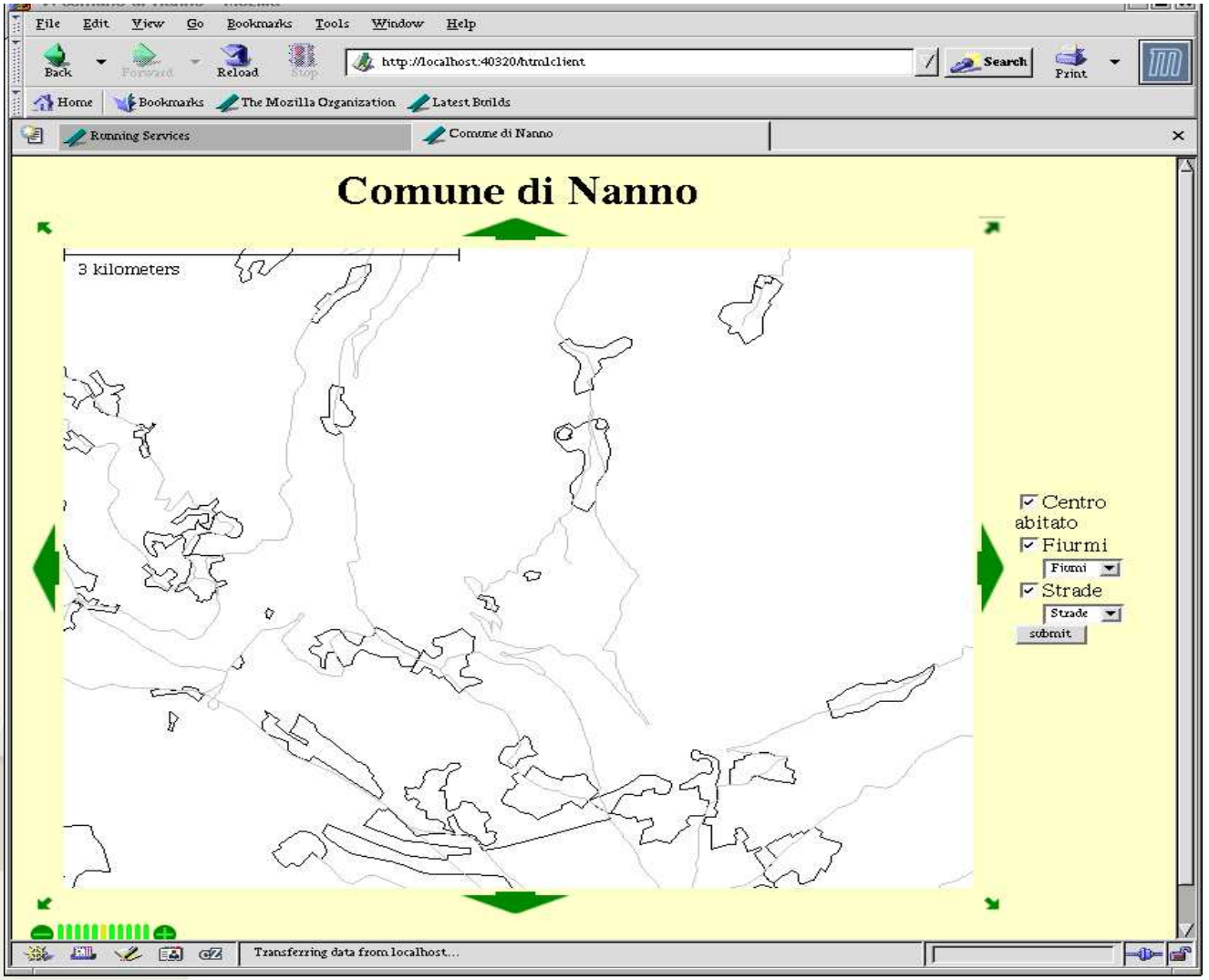
GISServer

```
java -Xms32m -Xmx600m gistoolkit.server.mapservice.WebMapService server.xml
```

http://localhost:43320/admin

Uso client side di GISServer

http://localhost:43320/htmlclient



Struttura del package

Gistoolkit

- **features** — package con degli oggetti di base come: Point, Line, Poligon ... più un insieme di tool per gestirli.
- **common** — package di supporto per la gestione di file XML di configurazione.
- **config** — package per la lettura/scrittura dei file XML di configurazione
- **projection** — package per la gestione dei sistemi di proiezione
- **datasources** — package per la gestione delle sorgenti dati sia in lettura che scrittura
- **display** — package per la visualizzazione dei dati
- **application** — applicazione di esempio GISEditor
- **server** — applicazione di esempio GISServer

Package datasources

Gestione delle fonti dati

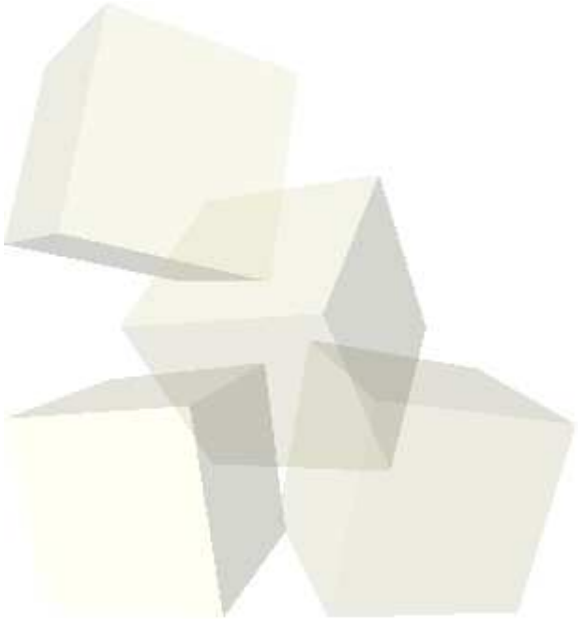
- Il package contiene una buona varietà di fonti dati
- Ogni tipo di fonte è gestita da un sotto-package
- Le attuali datasources disponibili sono:
  - ESRI Shape file
  - Image file
    - tiff+tfw
    - jpg+jfw
    - I formati di sopra a meno del world file aggiungendo le coordinate
  - ARCSDE
  - ARCims
  - IBM DB2 Spatial Extender
  - OGCWebService
  - PostGIS
  - Terraserver

Package `datasources arcsde/arcims`

Uniche datasources basate su software proprietario

package
`proprietary.datasources.arcsde`
`proprietary.datasources.arcims`

Per essere usate occorre avere una licenza delle classi ESRI per ArcSDE e ArcIMS



Package datasources.shapefile

Esempio

```
import gistoolkit.datasources.shapefile.*

public class TestShapeFileDataSource {
    ...
    public ShapeFileDataSource getShapeFileDataSource(String
fileName, String path) {
        ShapeFileDataSource myShapeFileDataSource = null;
        try {
                myShapeFileDataSource = new ShapeFileDataSource(path
+ File.separator() + fileName);
        } catch (Exception e) {
                System.out.println("Error " + e);
        }
        return (myShapeFileDataSource);
    }
    ...
}
```

Package datasources.imagefile

Gestione dati raster (tiff, jpg, bmp)

Basato sulle classi JAI (Java Advanced Imaging) di SUN
Carica in automatico il World File associato (se presente)
In alternativa è sempre possibile dichiarare la posizione, secondo il sistema di riferimento, inserendo i valori
Per rendere più efficente l'uso dei raster vengono proposte due utility:

ImageChopper

tool che divide in 4 parti un raster

RasterCatalog

tool che genera un ramo di directory dai raster di origine a diverse scale e in diversi 'tile'

Entrambi le utility generano un file XML che verrà poi gestito dalla corrispettiva datasource

Il risultato è una ottimizzazione della gestione dei raster

Package `datasources.postgis`

Gestione db spatial extension di PostgreSQL

Si divide in due datasource

ReadOnlyPostGISDataSource

accede ai dati in sola lettura

UpdateblePostGISDataSource

permette la modifica dei dati

Problema sulla implementazione:

ogni volta che viene istanziata una PostGISDataSource viene aperta una connessione al database

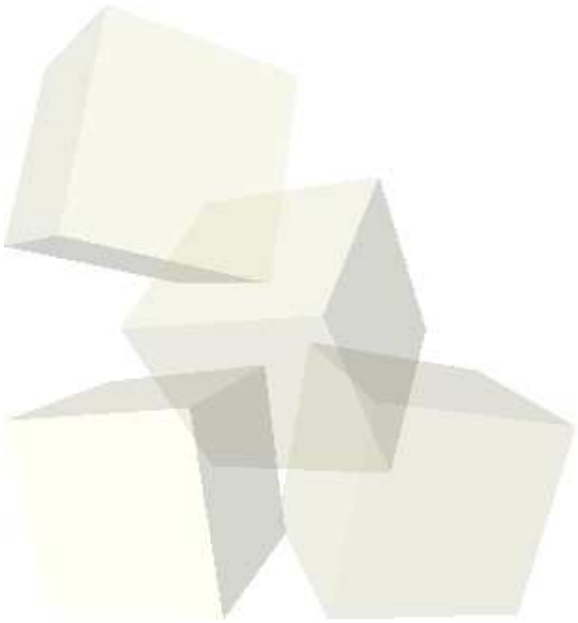
Soluzioni al problema:

rendere la classe `static`, instanziarla una sola volta nel codice e cambiarne gli attributi attraverso i vari metodi a disposizione

Dai dati all'analisi

Una volta caricata una sorgente dati è possibile cominciare l'analisi dei dati utilizzando la classe `GISDataSet` contenuta nel package `datasources` e da lì accedere alle varie features (Record, Attribute, Shape, Envelope ...) in essa contenute

```
...
   DataSource dataSource = new dataSource();
   GISDataSet dataSet = dataSource.readDataset();
...
```



Package features.GISDataSet

Astraendo il GISDataSet descrive una tabella

Attributi →	Attribute[0]	Attribute[1]	Attribute[2]
Record[0] →	Trento	100.000	C
Record[1] →	Rovereto	30.000	C
Record[2] →	Pergine	15.000	C

gistoolkit.features.Record

Features per le analisi spaziali

```
...
DataSource dataSource = new dataSource();
GISDataSet dataSet = dataSource.readDataset();
Envelope envelope = dataSet.getEnvelope();
// Solo su DataSource di tipo vettoriale
Shape[] shapes = dataSet.getShapes();
...
```

gistoolkit.features.Envelope

Rappresenta un'area rettangolare che include l'intero tematismo (Layer)

gistoolkit.features.Shape

Rappresenta una singola geometria (`Point, LineString, Poygon...`)

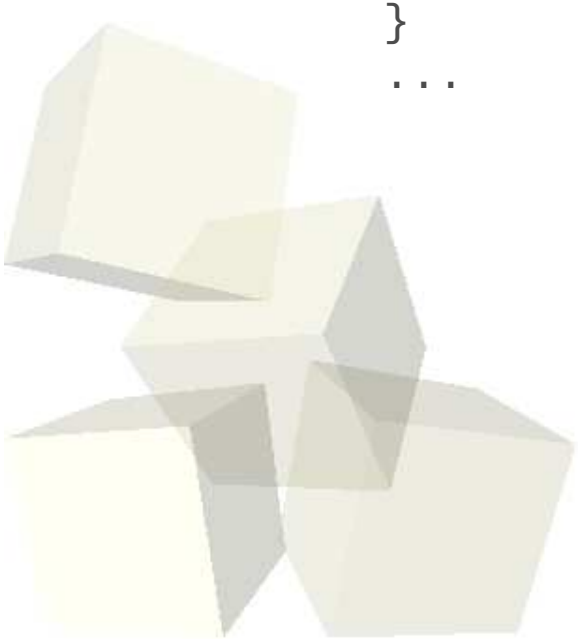
Si può ricavare anche dall'oggetto Record

```
...
Record record = dataSet.getRecord(0);
Shape shape = dataSet.getShape
System.out.println(shape.getShapeType());
...
```

Es. di procedura di una analisi spaziale

```
...
DataSource dataSource = new dataSource();
GISDataSet dataSet = dataSource.readDataset();
Envelope envelope = dataSet.getEnvelope();
// Solo su DataSource di tipo vettoriale
Shape[] shapes = dataSet.getShapes();

if (shapes[0].intesects(shapes[25])) {
      System.out.println("Il record 0 interseca il record
25");
}
...
```



Java Topology Suite

API java distribuite con licenza LGPL da parte della VIVID Solutions in grado di fare analisi complesse su dati vettoriali spaziali.

Attualmente è in atto un porting di queste classi in C++ (progetto GEOS) per potenziare il core di PostGIS

La struttura dei dati deriva dalle specifiche del Open Gis Consortium.
In particolare sulla struttura WKT (Well Know Text)

JTS e GISToolkit

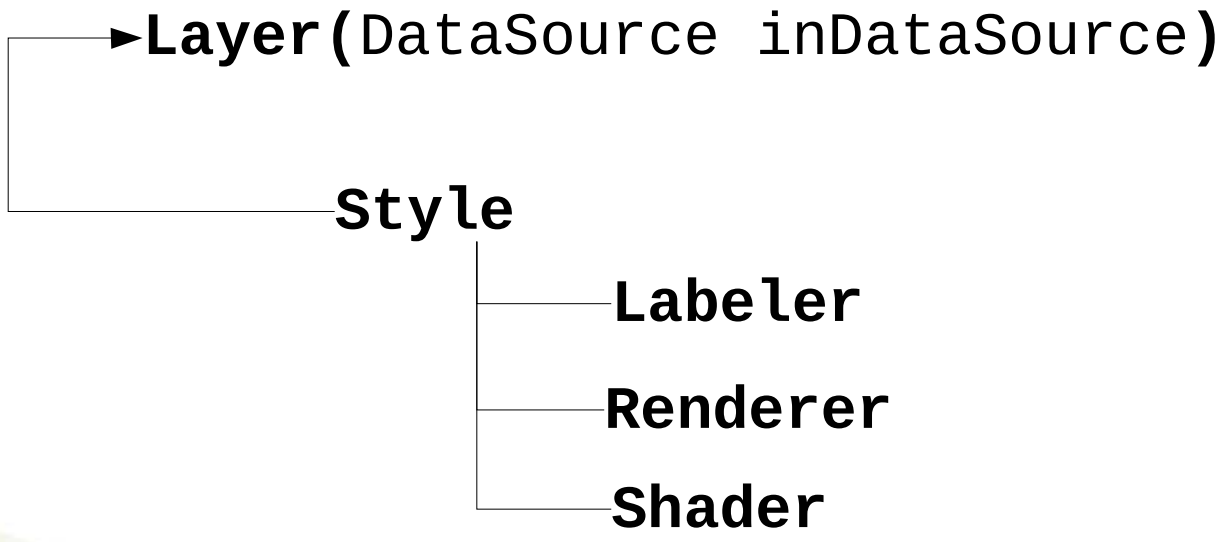
Attualmente è in corso lo sviluppo di una classe (package features) per l'integrazione con GISToolkit.

In attesa occore procedere secondo questo schema:

- utilizzare il metodo `.getWKT()` da un oggetto Shape (che restituisce una stringa)
- usarlo nei costruttori delle Geometry di JTS
- procedere con i metodi delle Geometry di JTS
- recuperare la stringa WKT dell'oggetto JTS
- usare il metodo `getShape()` dalla datsource statica `gistoolkit.datasources.WKTDataSource`

Visualizzazione dei dati

```
Package gistoolkit.display.*
```



Per una completa overview sull'uso degli stili dei layer è sufficente eseguire l'applicazione di esempio GISEditor dove il nome di ogni comando corrisponde al nome della classe che verrà utilizzata

Creazione applicazioni DESKTOP

Package gistoolkit.display.*

GISDisplay

Estende la classe JPanel delle classi swing di SUN
Una volta creata una classe che estende la classe Jframe è sufficente istanziare l'oggetto GISDisplay per avere una 'area mappa' ed usare il metodo addLayer() per visualizzare i dati

...

```
public class GisToolKitGUI extends JFrame {
   public myGISDisplay;

    public void initPanel() {
        myGISDisplay=new GISDisplay();
        myGISDisplay.setBackground(Color.white);
        Container myPanel = this.getContentPane();
        myPanel.add(myGISDisplay);
    }

    public void addLayer(Layer inLayer) {
        myGISDisplay.add(inLayer)
    }
```

La navigazione della mappa avviene cambiando l'Envelope del GISDisplay

Sviluppo di applicazioni server

Nelle soluzioni webgis si usano principalente due metodi:

- generazione di bitmap server side su richiesta dell'utente tenendo in sessione l'ultima vista

- creazione di vettoriali gestibili da un browser

La soluzione avviene utilizzando il metodo .drawLayer() della classe Layer.
Questo metodo richiede due oggeti:

Graphics (delle AWT di sun)

Converter (oggetto necessario a convertire le coordinate dello schermo nelle coordinate del sistema di riferimento usato)

Chiaramente, qualora sia necessario visualizzare più Layer occorre ripetere l'operazione di 'draw'per ciascuno sempre sugli stessi oggetti

La classe Converter

Package gistoolkit.display.Converter

Il costruttore della classe richiede due Envelope:
- uno espresso in coordinate screen
- l'altro espresso in coordinate del sistema di rifermento

L'utente, client side, opera sulla mappa facendo uso di un sistema di coordinate in pixel (prendendo come riferimento l'immagine che viene presentata)

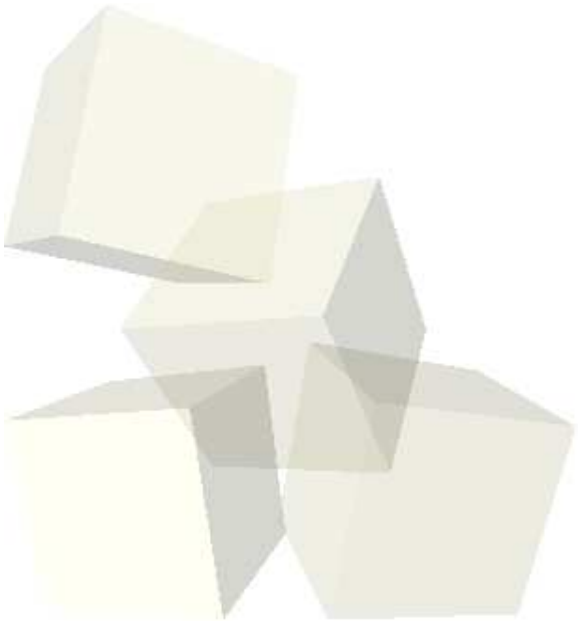
Tramite l'oggetto Converter e tenendo l'ultimo Envelope reale spedito al client, si possono recuperare le infomazioni sulla nuova vista da proporre.

La navigazione avviene inserendo l'Envelope desiderato a tutti i Layer caricati

Generazione delle immagini

La generazione delle immagini avviene tramite l'uso dell'oggetto Graphics e delle classi JAI

Ottenuta una BufferedImage è possibile inviare l'output su o su un file o sull'output di default di un servlet contaner (es. Apache Tomcat)



SVG Scalable Vector Graphics

Informazioni riguardo l'uso di GIS e SVG

http://www.carto.net

Il progetto Apache Batik si offre API per la creazione e gestione del formato SVG.

Fra le classi disponibili

SVG2DGraphics

che estende l'oggetto Graphics2D di SUN (a sua volta 'padre' di Graphics)

Per generare un file SVG con GISToolkit basta utilizzare questo oggetto, al posto dell'oggetto Graphics, come argomento del metodo '.drawLayer()' della classe Layer

Punti a sfavore:

- mancato supporto per i sistemi di riferimento
- API non complete per l'uso di ECMAScript

Conclusione

GISToolKit
http://gistoolkit.sourceforge.net

Open Gis Consortium
http://www.opengis.org

Java Topology Suite
http://www.vividsolutions.com/jts/jts_frame.htm

PostGIS
http://postgis.refractions.net

FreeGIS - portale sul GIS e il software libero
http://www.freegis.org

GNU
http://www.gnu.org