

---

# Web GIS

Claudio Rocchini  
Istituto Geografico Militare

# Introduzione

---

- La cartografia su web sta diventando il maggior sistema di divulgazione dei sistemi GIS
- La tecnologia ha dovuto trovare la sua strada a fatica fra le pieghe dei prodotti commerciali, standard di fatto o teorici, lotte di potere (es. applet java e Microsoft).

# Scelte Tecniche

---

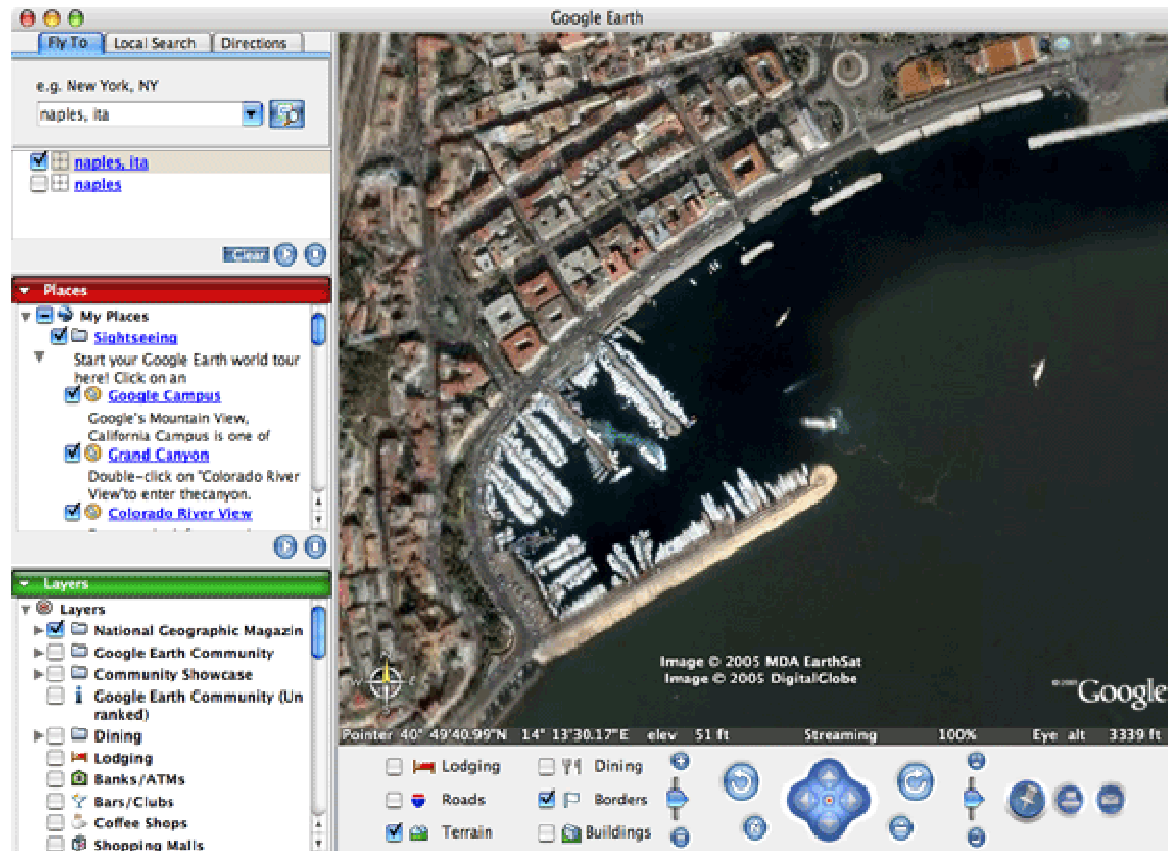
- Le scelte tecniche da adottare derivano da una serie di opzioni:
  - Tecnologia a livello client
  - Tecnologia a livello server
  - Tipologia di Dati (Raster, TIN)
  - Freq. Aggiornamento dei dati (Warehouse, Sistema a transazioni).

# Tipologie di Client

---

- La prima scelta è sulla tecnologia del client:
  - Applicazione dedicata con connessione di rete (es. Google Heart, NASA World Wind)
  - Pagina web con plug-in (es. java applet, flash, etc)
  - Pagina web html “pura” (Google Heart, Catalogo Interattivo IGM).

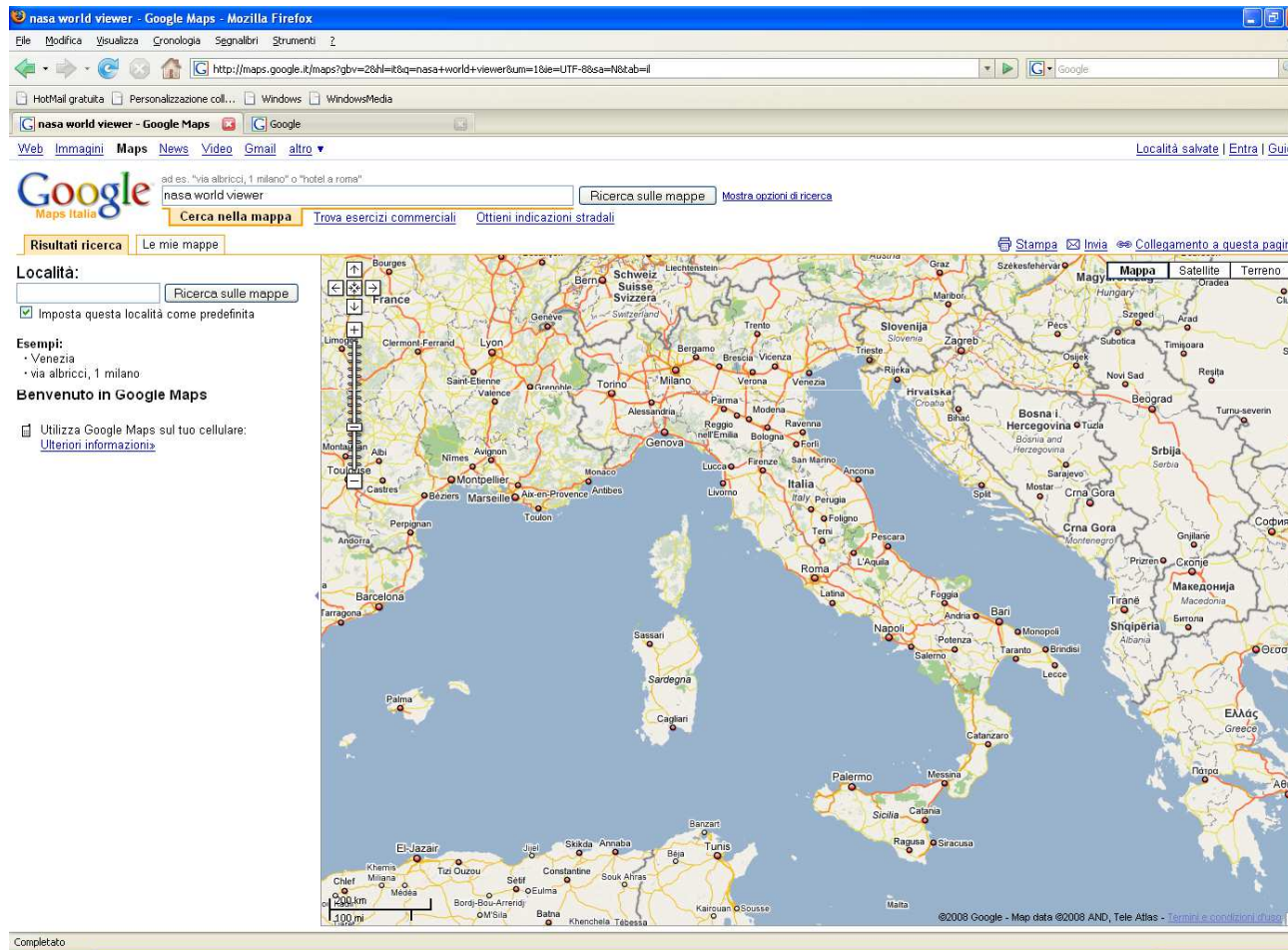
# Google Heart



# NASA World Wind



# Google Mappe



# Client Dedicato

---

- Pro
  - Massima Flessibilità nell'interfaccia
  - Massima potenzialità di utilizzo
- Contro
  - Necessita installazione software
  - Necessita aggiornamento
  - Non facilmente integrabile con altri servizi web o con GIS
  - Non portabile (es. su Macintosh e Linux)
  - Produzione del dato: impossibile o soggetta ad acquisto di software proprietario



# Pagina Web + Plug-in

---

- Pro
  - Massima flessibilità nell'interfaccia
  - Facilmente integrabile con altri servizi web
- Contro
  - Necessita installazione ed aggiornamento
  - Soggetto a lotte di potere (es. applet-java controllo Microsoft che ha deciso di boicottarlo)
  - Non integrabile con GIS
  - Produzione del dato: soggetta all'acquisto di software commerciale

# Pagina Web “Pura”

---

- Pro
  - Nessuna installazione o aggiornamento software
  - Nessun software necessario alla produzione del dato
  - Portabile su vari sistemi (Macintosh e Linux)
  - Integrabile con altri sistemi web
- Contro
  - Scarsa versatilità dell'interfaccia

# Note sul client

---

- Dal punto di vista di un produttore (o divulgatore) dati di media grandezza:
  - Client Dedicato: scelta impraticabile
  - Client + Plug-in: necessita di acquisto software di produzione del dato (o di divulgazione es. er-viewer ecw)
  - Client Html Puro: poca interattività dell'interfaccia.

# Interfaccia Web Pura: note

---

- Formato dei dati: possibile solo la visualizzazione di dati raster
- I dati vettoriali dovranno essere rasterizzati (a priori o al volo)
- Si sta tentando di lanciare il formato vettoriale SVG, Adobe lo supporta, Wikipedia lo sponsorizza, ma bisogna vedere...

# Realizzazione di un server Web GIS

---

- Primo Passo: il sistema operativo del server
- Windows
  - Non spaventa l'utilizzatore
  - Rallentamento generalizzato (es. chiamata cgi)
  - Scarso livello di sicurezza
  - Difficile manutenzione a distanza
- Linux
  - Veloce e sicuro
  - Manutenzione a distanza completa (telnet)
  - Pacchetti software già pronti
  - Es. Oracle consiglia linux come server

# Realizzazione di un server Web GIS

---

- Secondo passo: server web
- Il server web gestisce il sito nella sua struttura di base: pubblica le pagine statiche (testo ed immagini) ed esegue i linguaggi dinamici (cgi, php, perl, java)
- Apache (open source gratuito) è il leader in questo campo, tecnologicamente migliore di molti concorrenti (es. MS IIS)

# Realizzazione di un server Web GIS

---

- Terzo Passo: scelta della fonte dei dati
- Dati in formato nativo:
  - File shape
  - Raster (es. geotiff)
- Database Spaziale
  - La maggior parte dei dati (esclusi raster) è bene che stia in un db spaziale: Oracle o Postgres
- La buona strutturazione delle fonti di dati è la base del buon funzionamento del web.

# Realizzazione di un server Web GIS

---

- Quarto Passo: scelta del server cartografico
- Il server cartografico si occupa dell'interazione fra i web e le fonti di dati:
  - Gestisce la connessione alle fonti di dati
  - Rasterizza i dati vettoriali
  - Esegue le query spaziali



# Server Cartografici Commerciali

---

- Esistono molti prodotti commerciali:
  - ESRI ARCIMS
  - ESRI WEBMAP
  - **Intergraph GeoMedia® WebMap**
- Pro
  - Facile creazione di progetti da pubblicare attraverso le interfacce GIS
- Contro
  - Costo
  - Necessità di aggiornamento continuo
  - Bassa valenza tecnologica (quelli open source sono superiori dal punto di vista tecnologico)
  - Necessità di legarsi ad un particolare produttore

# Geoserver

---

- Open Source (gratuito)
- Scritto come servlet java (necessita installazione di TomCat o JBoss: si affiancano a apache per gestire applicazioni java)
- Potente e versatile
- Gestisce in modo completo i servizi WFS e WMS (ne parliamo dopo)
- Gestisce molti fonto di dati (raster, oracle, postgres)
- Sito: <http://geoserver.org>

# Geoserver

The screenshot shows the GeoServer Home page in a Mozilla Firefox browser. The browser's address bar displays the URL <http://geoserver.org/display/GSOS/GeoServer+Home>. The page features a navigation menu on the left with categories like Download, Documentation, Community, and RnD. The main content area includes a 'GeoServer Home' header, a red warning box stating 'This is the test instance of Confluence hosted by TOPP', and a 'What is Geoserver' section. This section explains that GeoServer is an Open Source server for geospatial data, supporting various protocols and formats. It also mentions that GeoServer is built on the Geotools Java toolkit. A prominent green 'Download Now Free' button is visible. To the right, there is an 'OGC Certified' section with a map of New York City streets and logos for OGC Certified, WMS 1.1.1, and WFS 1.0. Below this, there is a 'GeoServer Blog' section with links to 'GeoServer 1.6.0', 'Another GeoServer Roadmap update', and 'GeoServer 1.6.0-RC3 Released!'. The footer of the page contains the text 'Completato'.

# Mapserver

---

- Open Source (gratuito)
- Tecnologia CGI (non necessita di Tomcat, ma rallenta sui sistemi Windows, possibile utilizzo con fastcgi)
- Flessibile e Potente
- Gestisce molti fonto di dati (raster, oracle, postgres)
- Sito: <http://mapserver.gis.umn.edu/>

# Un esempio concreto: Warehouse punti trigonometrici

---

- Realizzata come:
- S.O.: Linux (Fedora 7 64bit)
- Apache
- Postgres (dati vettoriali)
- TIF + TFW tiled per dati raster
- Mapserver
- PHP: gestione html dinamico (query)
- LateX (generazione pdf dinamici)
- Interfaccia Dinamica Javascript

# Nota: controlli di accesso

---

- Interamente delegata a Postgres
- Di solito i db hanno un sistema di controllo di accesso molto sofisticato.
- L'accesso alla pagina web puo' essere delegato al database tramite l'utilizzo di sessioni PHP.
- Es: l'accesso alle coordinate è filtrato dal database:

# Filtro sulle coordinate

---

```
CREATE OR REPLACE VIEW live.coordview AS
  SELECT "Coordinate"."ID", "Coordinate"."Caposaldo",
         CASE
           WHEN "current_user"() = 'demo'::name THEN floor("Coordinate"."Roma40 FI" *
1000::double precision) / 1000::double precision
           ELSE "Coordinate"."Roma40 FI"
         END AS "Roma40 FI",
         CASE
           WHEN "current_user"() = 'demo'::name THEN floor("Coordinate"."Roma40 LA" *
1000::double precision) / 1000::double precision
           ELSE "Coordinate"."Roma40 LA"
         END AS "Roma40 LA",
         CASE
           WHEN "current_user"() = 'demo'::name THEN floor("Coordinate"."WGS84 FI" *
1000::double precision) / 1000::double precision
           ELSE "Coordinate"."WGS84 FI"
         END AS "WGS84 FI",
         CASE
           WHEN "current_user"() = 'demo'::name THEN floor("Coordinate"."WGS84 LA" *
1000::double precision) / 1000::double precision
           ELSE "Coordinate"."WGS84 LA"
         END AS "WGS84 LA"
  FROM live."Coordinate";
```

# Mapserver

---

- Utilizzo mapserver:
- Installazione inesistente (basta copiare il file del programma nella cartella cgi-bin di apache)
- Necessita di una serie di file di configurazione (.map)
- Due utizzi:
  - Gestione completa della pagina di mappa (ma molto grossolana)
  - Gestione della sola rasterizzazione, interfaccia realizzata con altri mezzi (javascript): scelta adottata.



# Esempio configurazione Mapserver: map

---

## MAP

```
STATUS ON
#DEBUG ON
NAME "Igm95"
EXTENT 6511228 3976444 7610228 5257944
IMAGETYPE pngrgb
UNITS meters
IMAGECOLOR 27 150 250

SYMBOLSET 'igm95.sym'

INCLUDE "raster_layer.map"
```

## LAYER

```
NAME "caposaldi"
DUMP true
TYPE point
PROCESSING "CLOSE_CONNECTION=DEFER"
CONNECTIONTYPE postgis
CONNECTION "user=demo password=demo
dbname=ware host=172.17.3.150
port=5432"
DATA "geom FROM live.funico"
MINSCALE 100
MAXSCALE 300000
CLASS
  SYMBOL 's_caposaldo'
  MINSCALE 100
  MAXSCALE 250000
END
CLASS
  SYMBOL 's_caposaldo_r'
  MINSCALE 250000
  MAXSCALE 300000
END
END
```

# Esempio configurazione Mapserver: map

---

```
LAYER
  NAME "igm5"
  DUMP true
  TYPE point
  PROCESSING "CLOSE_CONNECTION=DEFER"
  CONNECTIONTYPE postgis
  CONNECTION "user=demo password=demo
    dbname=ware host=172.17.3.150
    port=5432"
  DATA "geom FROM igm95.funico5"
  MINSCALE 100
  MAXSCALE 600000
  CLASS
    SYMBOL 's_5'
    MINSCALE 100
    MAXSCALE 250000
  END
  CLASS
    SYMBOL 's_5_r'
    MINSCALE 250000
    MAXSCALE 600000
  END
END

#OUTPUTFORMAT
# NAME aggrgb
# DRIVER AGG/PNG
# IMAGEMODE RGB
# EXTENSION "png"
# FORMATOPTION "INTERLACE=ON"
#END

OUTPUTFORMAT
  NAME pngrgb
  DRIVER "GD/PNG"
  MIMETYPE "image/png"
  IMAGEMODE RGB
  EXTENSION "png"
  FORMATOPTION "INTERLACE=ON"
END

END
```

# Esempio configurazione Mapserver: symbol

---

**SYMBOLSET**

```
SYMBOL
  NAME 's_trigo'
  TYPE pixmap
  IMAGE "icons/0.gif"
  TRANSPARENT 15
```

**END**

```
SYMBOL
  NAME 's_trigo_r'
  TYPE pixmap
  IMAGE "icons/0r.gif"
  TRANSPARENT 7
```

**END**

**SYMBOL**

```
NAME 's_caposaldo'
TYPE pixmap
IMAGE "icons/1.gif"
TRANSPARENT 63
```

**END**

**SYMBOL**

```
NAME 's_caposaldo_r'
TYPE pixmap
IMAGE "icons/1r.gif"
TRANSPARENT 31
```

**END**

# Esempio configurazione Mapserver: sfondo

---

```
LAYER
  STATUS default
  NAME "ras5M"
  TYPE raster
  DATA "raster/scala_5Mtiled.tif"
  MINSCALE 250000
  MAXSCALE 600000
END
```

```
LAYER
  STATUS default
  NAME "ras2e5M"
  TYPE raster
  DATA "raster/scala_2e5Mtiled.tif"
  MINSCALE 125000
  MAXSCALE 150000
END
```

```
LAYER
  STATUS default
  NAME "ras1M"
  TYPE raster
  DATA "raster/scala_1Mtiled.tif"
  MINSCALE 50000
  MAXSCALE 60000
END
```

```
LAYER
  STATUS default
  NAME "ras500M1"
  TYPE raster
  DATA
    "raster/scala_500K_1tiled.tif"
  MINSCALE 25000
  MAXSCALE 30000
END
```

# Configurazione per supporto al WMS/WFS

---

## MAP

```
STATUS ON
DEBUG ON
NAME "Igm95"
CONFIG "PROJ_LIB" "/usr/share/epsg_csv"
EXTENT 6511228 3976444 7610228 5257944
IMAGETYPE pngrgb
UNITS meters
IMAGECOLOR 27 150 250
```

```
SYMBOLSET 'igm95.sym'
```

```
#TODO; la proiezione e' sbagliata
```

## PROJECTION

```
"proj=utm"
"ellps=WGS84"
"datum=WGS84"
"zone=32"
"units=m"
"north"
"no_defs"
```

```
END
```

## WEB

```
LOG "/var/www/cgi-bin/mapserv.conf/mapserv.log"
```

## METADATA

```
"wms_title" "Igm95"
"wfs_title" "Igm95"
"wms_onlineresource"
"http://bedem.igm.mil/cgi-
bin/mapserv?MAP=mapserv.conf/igm95.map&"
"wfs_onlineresource"
"http://bedem.igm.mil/cgi-
bin/mapserv?MAP=mapserv.conf/igm95.map&"
"wms_abstract" "Warehouse punti trigonometrici
IGM"
"wms_srs" "EPSG:32632"
"wfs_abstract" "Warehouse punti trigonometrici
IGM"
"wfs_srs" "EPSG:32632"
```

```
END
```

```
END
```

# Configurazione per supporto al WMS/WFS

```
INCLUDE "raster_layerogc.map"

LAYER
  NAME "caposaldi"
  DUMP true
  TYPE point
  PROCESSING "CLOSE_CONNECTION=DEFER"
  CONNECTIONTYPE postgis
  CONNECTION "user=demo password=demo dbname=ware
    host=172.17.3.150 port=5432"
  DATA "geom FROM live.funico"
  MINSCALE 100
  MAXSCALE 300000
  CLASS
    SYMBOL 's_caposaldo'
    MINSCALE 100
    MAXSCALE 250000
  END
  CLASS
    SYMBOL 's_caposaldo_r'
    MINSCALE 250000
    MAXSCALE 300000
  END
  METADATA
    "wfs_title" "caposaldi"
    "gml_featureid" "caposaldi"
  END
END

LAYER
  NAME "igm5"
  DUMP true
  TYPE point
  PROCESSING "CLOSE_CONNECTION=DEFER"
  CONNECTIONTYPE postgis
  CONNECTION "user=demo password=demo dbname=ware
    host=172.17.3.150 port=5432"
  DATA "geom FROM igm95.funico5"
  MINSCALE 100
  MAXSCALE 600000
  CLASS
    SYMBOL 's_5'
    MINSCALE 100
    MAXSCALE 250000
  END
  CLASS
    SYMBOL 's_5_r'
    MINSCALE 250000
    MAXSCALE 600000
  END
  METADATA
    "wfs_title" "igm5"
    "gml_featureid" "igm5"
  END
END
```

# Parametri Variabili

---

- I parametri variabili di mapserver:
  - Layer da visualizzare
  - Layer da interrogare
  - Zona della mappa
  - Zoom corrente
  - ...
- Possono essere comunicati direttamente nella richiesta web al server.

# Performance

---

- Dati vettoriali
  - Gli indici spaziali devono essere creati correttamente
  - Ci possono essere più versioni dei dati vettoriali, adattati alle varie scale
  - Evitare cambiamenti di sistema di riferimento
  - Oggetti giganti possono essere suddivisi in pezzi (autostrada A1), altrimenti entrano sempre in gioco nelle query spaziali
- Raster
  - Suddivisione delle immagini giganti in griglie
  - Utilizzo di tif tiled (supportati da mapserver)
  - Riduzione al minimo dello spazio (migliore compressione o riduzioni colori, scelta del formato in rapporto qualità/performance)



# Qualità della resa grafica

---

- Se gli strati dello sfondo sono fissi si possono pre-rasterizzare utilizzando software dedicato (nel nostro caso arcgis)
- Possibilità di attivare l'utilizzo della libreria AGG-Graphics (rasterizzazione con anti-aliasing)

# Interfaccia Dinamica Html “Pura”

---

- Javascript: linguaggio di programmazione utilizzabile in locale, sul client delle pagine web.
- Tramite questo linguaggio è possibile manipolare in modo dinamico il contenuto della mappa.
- Es: possibile costruire il panning dinamico, lo zoom, l'interrogazione della mappa.
- Possibile creazione di overlay sulla mappa (indirizzi di Google Mappe, scala dello zoom)

# Esempio di codice Javascript

---

```
function sethome() {
    tileidx = stileidx;
    tileidy = stileidy;
    maposx  = smaposx;
    maposy  = smaposy;
    curzoom = 0;
    pixsiz  = pixsizv[curzoom];

    for(i=0;i<ntilet;++i)
        images[i].style.visibility = 'hidden';

    set_position();
    set_source(true);
}
```

# Esempi di Interfacce Dinamiche

---

- Google Mappe (studiabile)
- Ka-Map: pacchetto open source preconfigurato
- IGM: sviluppo interno del sistema, due versioni:
  - Stand-Alone con sorgenti dati raster (catalogo interattivo su [igmi.org](http://igmi.org))
  - Collegato a mapserver: catalogo [igm95](#)

# Ajax

---

- Ulteriore tecnologia di supporto alle interfacce web (non utilizzata in IGM)
- Possibilità tramite javascript di eseguire richieste asincrone e parallele di dati:
  - Caricamento parallelo di più tile contemporaneamente
  - Caricamento in background delle tile adiacenti
- Svantaggi
  - Non completamente standardizzato fra i vari visualizzatori (IE Explorer varie versioni, Firefox, Mosaic, Etc)
  - Richiede continuo aggiornamento per i nuovi visualizzatori e grande sforzo di sviluppo.