

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# New Frontiers for WebGIS Platforms Generation

Davide Di Pasquale<sup>1</sup>, Giuseppe Fresta<sup>2</sup>, Nicola Maiellaro<sup>1</sup>,  
Marco Padula<sup>1</sup> and Paolo Luigi Scala<sup>1</sup>

<sup>1</sup>*ITC-CNR, Construction Technologies Institute, Italian National Research Council*

<sup>2</sup>*ISTI-CNR, Institute of Information Science and Technology "Alessandro Faedo",  
Italian National Research Council  
Italy*

## 1. Introduction

Information intensive applications usually involve highly collaborative activities and aspects throughout all the different stages of the information seeking and retrieval process. They are usually performed by work groups composed by heterogeneous professionals: a pervasive collaboration of diverse partners is needed to harmonize different tasks through the common objective.

Team components are usually located in different physical structures and the workflow tasks, such as data harvesting, aggregation, elaboration and presentation, need a high quality and quantity of communication to deliver a result with high standard level. Designing and implementing an integrated system able to deal with large amount of highly heterogeneous data, to allow distributed data access and to provide collaboration between partners requires a deep analysis of the diverse requirements and habits of all the operators.

Such a development environment should take into account all these requirements and should be open to real time modifications and improvements with the audit of the final users.

Target domains for such kind of information systems are environment heritage conservation and tourism promotion: these domains refer to a complex production chain where private companies and public institutions are involved with specific interests and skills.

A relevant amount of the information needed and produced in these activities are referred to the territory, therefore have to be represented as geo-referenced data and the system being set up in order to create and share each task of the knowledge workflow must support a widespread typology of data (Barricelli et al., 2008).

Two important aspects are to be stressed in the design of this kind of information systems: one is the possibility to shorten the publication cycle of the information, providing authoring tools to the information providers in order to let them quickly edit and/or update information with a minor help from the IT staff. The other is the ability of creating a customizable environment that can be adapted to the diverse professionals involved in the whole process.

Such a complex system can nowadays be developed by integrating already available components and making them inter-operable or adopting an integrated approach and developing a single integrated design environment. The latter approach has many benefits in terms of homogeneity in design and usability both for content generator actors and end users.

In this work we present a proposal for the creation of an integrated design and content management environment for a WebGIS application. The proposed system is addressed to local administrations in collaboration with diverse data providers and should be designed implementing a methodology that supports a strict cooperation between developers and end users in terms of features' design and use feedback.

To address these requirements, the Software Shaping Workshop methodology for the design and development of virtual environments has been adopted, as described in Section 3: the proposed architecture will support all the stakeholders involved in the environment heritage conservation and tourism promotion workflow by implementing information manipulation and exploitation functionalities on three levels: meta-design level, design level and use level.

The fairly extended state of the art presented in Section 2 highlights the novelty of our approach by offering an integrated environment covering the complete workflow, realized by the means of a network of specialized sub-environments where all the actors can exchange and produce information that will be stored in a shared knowledge base.

## **2. The state of the art**

In the last decades, the diffusion of Internet access and the development of the Web is allowing easy and fast access to information for an increasing number of users. Spatial (or geographic) information is a kind of information with increasing importance nowadays: solving spatially oriented problems in order to make decisions in both public and private sector is a consolidated emerging trend in ICT. The need to combine data from various sources in a consistent and high quality way assumes readability of data and further knowledge of data origin, quality, compatibility and structure: accepting and promoting standards and norms for digital data and computer based communication processes affects the sustainability of digital content as well as the compatibility of data, software and hardware.

### **2.1 Standards for data exchange in geomatics applications**

Developing and adopting joined approaches and standards for spatial data acquisition and exchange provides many benefits like portability, interoperability, and maintainability. Standards organizations are active at multiple levels, such as government organizations, e.g. The American National Standards Institute (ANSI), the International Organization for Standardization (ISO), and industry associations, such as the Open Geospatial Consortium (OGC, formerly known as OpenGIS) (Groot, McLaughlin, 2000). The OGC Abstract Specification, for example, defines all OGC Web Services (OWS) that follow the Service Architecture Interoperability approach and provide models for metadata documentation; the Geography Markup Language (GML) specification models spatial features and

topological relationships between them and The Web Feature Service (WFS) describes a service-based supply of vector information as feature collections.

In addition to formal standards bodies and specification programs, there exist numerous de facto standards, like, for example, the file format widely used in the spatial data domain: the shapefile (by Environmental Systems Research Institute, ESRI (ESRI, a)). Other examples of de facto standards in matter of geo-data manipulation and exchange are:

- GeoRSS - Implements the approach of Really Simple Syndication (RSS) feeds. Applications such as Google Maps, Yahoo Maps and others implement GeoRSS since it has been released as an OGC White Paper in 2006 (Open Geospatial Consortium, 2006a).
- GeoJSON (GDAL) - Extends JavaScript Object Notation (JSON) to encode objects with location information expressing a variety of geographic data structures and thus JSON tools can also be used for processing GeoJSON data. Its format is human-readable and generally more compact than XML. Spatial data format types supported in GeoJSON include points, polygons, multi-polygons, features, geometry collections, and bounding boxes, which are stored along with feature information and attributes. GeoJSON is supported by numerous mapping and GIS software packages, including OpenLayers, GeoServer (GeoServer, 2008), GeoDjango (Django Software Foundation, 2005), GDAL (GDAL) and Safe Software FME (Safe Software) and also with PostGIS and Mapnik (Pavlenko, 2009) via the GDAL OGR conversion library. Yahoo! and Google also support GeoJSON in their API services.
- Tiled Maps - This approach implements a tile of small images loaded on request as the user pan the view, optimizing web traffic through cache exploitation (Open Source Geospatial Foundation, 2006). The very responsive feedback in map panning and zooming resulted in a widespread adoption of the Tiled Map Service Specification (TMSS), and WMS Tiling Client Recommendation (TCR).
- GML - The Geography Markup Language is “an XML grammar written in XML Schema for the modeling, transport, and storage of geographic information including both the spatial and non-spatial properties of geographic features” (Open Geospatial Consortium, 2006b). Following the OGC Abstract Specification data model, GML 3.2.1 models the reality in terms of features with complex, nonlinear, 3D geometry, features with 2D topology, features with temporal properties, dynamic features, coverage, and observations. It conforms to many ISO standards and, like XML, GML is only concerned with the representation of geographic data content. The representation of the geospatial features in the form of maps, can be obtained transforming GML data through Extensible Style sheet Language Transformations (XSLT).
- GDAL - Geospatial Data Abstraction Library is a translator library for geospatial raster data formats. It provides a single abstract data model to the calling application for all supported data formats (GDAL, 2011a).
- OGR Simple Feature Library is a C++ Open Source library, similar to GDAL, providing access to a variety of vector file formats. Most important supported formats are ESRI Shapefiles, SDTS, PostGIS, MapInfo mid/mif and TAB formats (GDAL, 2011b).
- SVG - Scalable Vector Graphics is an XML-based language that describes images in terms of simple elements, dots, poly-lines, elementary shapes. It is a standard developed by the W3C (World Wide Web Consortium) and uses mathematical

statements to describe the shapes and paths of an image so that can easily be made scalable. It features searchable text information and, being based on XML, this format conforms to other XML-based standards such as XML Namespace, XLink, and XPointer for linking from within SVG files to other files on the Web.

- **WFS** - The Web Feature Service Implementation Specification (Open Geospatial Consortium, 2005) has been developed by the OGC to allow the execution of queries and the extraction of features from the geographical data to respond to user requests. Its implementation allows a client to retrieve geospatial data encoded in GML from multiple Web Feature Services. The WFS is XML compliant and can be interfaced with diverse datastore technologies, uses XML through HTTP to communicate with clients and provides data manipulation operations on GML features.

## 2.2 Desktop GIS

Spatial information requires for its treatment and analysis special software tools, namely geographic information systems (GIS) (Konecny, 2003), (Longley, 2001). Common applications of these tools can be found in many sectors like land management, urban planning, and public administration, as well as many personal uses like leisure planning or social experiences sharing, so that GIS approach to spatial information has become an important part of a variety of information systems used for supporting decision-making processes in business, public administration, and personal matters. Geographic information system management tools have been traditionally developed as desktop standalone applications, due to the computing intensive activity typical of their role. Among the most significant Open Source software applications, at least the following should be surveyed.

**gvSIG:** a local and remote map viewer (2D and 3D) with support for the OGC standards, as well as a tool for publishing maps on paper and in map servers. It has a large variety of vector and raster analysis tools and enables the viewing and editing of maps in the field using palmtops or smartphones.

**GRASS:** developed for Unix platform, it's a mature software and since Quantum GIS (QGis) introduced an interface for GRASS into its functionalities, the diffusion of this software has increased even among non professionals. Its renewed 3D engine with support for vector geometries and raster data introduced a better managing of volumetric pixel (voxels) in 3D rendering. Grass features APIs for Python, Perl and PHP.

**MapWindow** (MapWindow Open Source Team) is developed for the .Net platform for Windows and it also features an ActiveX control. It has been adopted by the United States Environmental Protection Agency as the primary GIS platform and features a number of available plug-ins to expand compatibility and functionality. Current stable release is 4.7.

**Quantum GIS** or QGis (Open Source Geospatial Foundation, g) is a user friendly Open Source software that runs on Linux, Unix, Mac OSX, and Windows and supports vector, raster and database formats. QGIS is licensed under the GNU Public and is coded upon the QT library by Nokia. It supports PostGIS and SpatiaLite, the OGR library, ESRI shapefiles, MapInfo, SDTS and GML, the GDAL library, GRASS locations and mapsets and can manage all OGC-compliant services like WMS, WMS-C (Tile cache), WFS and WFS-T.



The system for Automated Geoscientific Analyses, or **SAGA** (Böhner) is a GIS for Windows and Linux with analysis tools for raster data, digital elevation models and numerical simulations like prediction of soil properties, terrain dynamics and climate parameters evolution.

**Open JUMP** (Jump Pilot Project) is a GIS application written in Java. It supports the GML format and the WMS protocol. its particular strength is the editing of geometry and attribute data. A growing number of vector analysis tools for topologic analysis and overlay operations is available. Current release 1.4 adds advanced raster data processing capabilities.

**OSSIM** (Open Source Geospatial Foundation, f) is an open source software with advanced geo-spatial image processing for remote sensing, photogrammetry, and GIS applications through a C++ library.

**OrbisGIS** (IRSTV) focuses on data acquisition techniques (remote sensing, modeling and simulation, site enquiries...), spatial data processing and representation (storage, modeling, multiscale 3D+t simulations) and is based on libraries such as JTS (Java Topology Suite) or ImageJ. It provides the ability to visualize or process 2D vector and/or raster data that may be stored in a flat file or remote DataBase Management System. Current version is 3.0

**uDig** (Refraction Research) is Java open source desktop application built with Eclipse Rich Client (RCP) technology. It can be extended with RCP “plug-ins” and vice-versa can be used as a plug-in in an existing RCP application. Version 1.2.2 is its current stable release.

Among **commercial desktop GIS** the most diffuse software are:

**ArcView** (ESRI, c), by ESRI; its key features are map authoring with templates, spatial query with query-building tools, basic modeling and analysis with custom reports generation, simple feature editing and data integration.

**Bentley Map** (Bentley Systems Inc., a) natively manages Oracle Spatial to store and edit all types of spatial data, can analyze 2D and 3D spatial data, supports many formats, features models and rule-based symbology and annotation, generates 3D scenes and animations.

**ERDAS** (ERDA Suite) presents a complete suite for turning acquired imagery into GIS data. It supports many source data formats, including orthos, terrain, features, maps, 3D data, land cover data and processing models.

Intergraph’s **GeoMedia** (GeoMedia) products suite is a set of integrated applications that provide access to geospatial data in many formats and bring an integrated geospatial view together, along with a set of analytic and editing tools.

## 2.3 Web GIS

Classic GIS software packages (desktop or professional GIS) have some drawbacks which limit their diffusion among all the users who need to use spatial information: first of all, the high costs; desktop software is then accessible only from the computer on which it is installed and their user interfaces requires training. The fact that desktop GIS is still a proprietary technology limits also the possible customization of their features. These problems along with spreading of the Internet and increasing demand for spatial

information have driven a rapid process of geo-enabling the Web and a rapid development of Internet GIS applications or Web GIS. Components of a typical Web GIS system include:

#### Data

- Spatial data – data with a positional or geographic component (maps), in some data file format (e.g. SHP, DWG, SDF, DGN) or stored in a spatial database (e.g. PostGIS, MySQL, Oracle Spatial, SDE)
- Attribute data – characteristics or properties of map features, stored as textual or tabular data, typically in a relational database

#### Software

- Map Server - the Web GIS core server application
- server middleware - to interpret requests from clients, interact with the web GIS application, and package the data for transfer via the web: often is integrated in the web GIS application
- web GIS Application – presentation layer application that interacts with the users and exposes the services of the Map Server. It's usually a thin or light client application that runs in the web browser and is based on client-side scripting languages.
- Web server – e.g. Apache, Internet Information Server
- Client web browser – e.g. Internet Explorer, Mozilla
- Client-side applet or plug-in – requirement depends on the technology
- Web-database application software – e.g. PHP, ASP.NET, ColdFusion

#### Hardware

- Central server computer
- Client computers
- Connection through the Internet or, for intranet sites, through a LAN or WAN

Some implementation examples of the main aspects of web GIS development are hereafter described.

Many **geospatial data**, and related metadata, are stored in database formats for more efficient accessing methods. With these tools the query for spatial characteristics is enabled and the ability to query them from another application is supported by the Open DataBase Communication (ODBC) standard. The most diffuse platforms are hereafter mentioned.

**MySQL** (SUN Microsystems, a) is the most used free (but not fully open source) database in Web applications and has recently introduced ad hoc *spatial extensions* (SUN Microsystems, b) that not yet follow the OGC Simple Features Interface Standard (SFS) (Open Geospatial Consortium) but allow to add geo-related capabilities to Structured Query Language (SQL).

**PostGIS**, the module for the Open Source Database PostgreSQL (PostgreSQL Global Development Group), provides storing capabilities and geographic analysis operations for geospatial information and offers a wide range of both free and proprietary tools, being also supported by ArcSDE, the of ESRI database access middleware and being fully compliant to the OGC SFS.

**pgRouting** extends the PostGIS / PostgreSQL geospatial database to provide geospatial routing functionality. It is the best option for network calculations and to analyse graphs; routes are processed directly in SQL, without using middleware.

Among proprietary and commercial software the most diffuse database with embedded geo-capabilities are:

**Oracle 11g**, with its extension Oracle Spatial (Oracle), provides tools for complex geospatial applications that require spatial analysis and processing. It includes full 3-D and Web services support to manage all geospatial data including vector and raster data, topology, and network models and is supported by the GDAL open source tool.

**ArcSDE** (ESRI, b), by ESRI, is a core component of ArcGIS for Server. It manages spatial data in a relational database management system (RDBMS) and enables it to be accessed by ArcGIS clients. The geodatabase is the primary data storage model for ArcGIS; it provides a single central location to access and manage spatial data.

Smallworld Technology's database technology called Version Managed Data Store (**VMDS**) has been designed and optimized for storing and analyzing complex spatial and topological data and networks typically used by enterprise utilities such as power distribution and telecommunications. The native Smallworld datastore can be stored in an Oracle Database. This allows the use of Oracle facilities for backups and recovery.

The core application in web GIS systems is the **map server** that, after having processed user requests, generates the requested map using data retrieved from spatial databases.

Due to the fact that commercial products are expensive, complex, and often not standards compliant, a number of Open Source Projects have reached high diffusion and high quality levels.

The main project is the map server of the University of Minnesota, **UMN MapServer** (University of Minnesota); born as a bunch of scripts to dynamically generate web maps from a ArcINFO GIS, it was developed by the North American Space Agency, the University of Minnesota and the Department of Natural Resources of Minnesota and is now a project of OSGeo maintained by tens of round-the-world developers. It can run as an executable Common Gateway Interface (CGI) or as a library and its main characteristics are: scale dependent feature drawing and application execution, feature labeling including label collision mediation, fully customizable template driven output, use of TrueType fonts, Map element automation (scalebar, reference map, and legend), Thematic mapping using logical or regular expression-based classes, support for popular scripting and development environments (e.g. PHP, Python, Perl, Ruby, Java, and .NET), cross-platform support, support of numerous Open Geospatial Consortium OGC standards (WMS, non-transactional WFS, WMC, WCS, Filter Encoding, SLD, GML, SOS, OM), a multitude of raster and vector data formats (TIFF/GeoTIFF, EPPL7), and many others via GDAL (ESRI shapefiles, PostGIS, ESRI ArcSDE, Oracle Spatial, MySQL) and many others via OGR.

The **deegree2** (Open Source Geospatial Foundation, a) open source map server is based on Java components and offers an high level of compliance to the OGC standards specifications (WMS, WFS-T, WCS, CSW, WPS and SOS). It is released under Lesser GNU Public License (L-GPL) and its main characteristics are: Simplified installation and configuration, Tool-based configuration (for WFS and WCS), Support of GML 3.1 with a complex Feature Model and 3D-geometries, Support of PostGIS 1.0 and Oracle spatial/locator (9i/10g), Advanced capabilities for object-relational mappings in the WFS, Multiple data sources for WMS



layers, Dynamic rendering rules within SLD, High-quality and large-size print outputs through Web Map Print Service (WMPS).

**GeoServer** is developed on the Java 2 Enterprise Edition (J2EE) platform. This allows the applications to be deployed on any J2EE compliant application server (Apache Tomcat, RedHat JBoss and Apache Geronimo, WebLogic and IBM WebSphere. It runs also as a middleware for geographic editing applications, supporting the transactional Web Feature Service (WFS-T) OGC protocol. It natively support the integration with OpenLayers (see next paragraphs).

In 2006 Autodesk released sources of its map server, **MapGuide Open Source**.

It includes client application that support feature selection, property inspection, map tips, and operations such as buffer, select within, and measure. MapGuide includes an XML database for managing content, and supports most popular geospatial file formats, databases, and standards. It can be deployed on Linux or Windows, supports Apache and IIS web servers, and offers extensive PHP, .NET, Java, and JavaScript APIs for application development as well as tools for AutoCAD publication. MapGuide Open Source is licensed under the LGPL.

**TileCache** (MetaCarta) is indeed a Python-based WMS-C/TMS middleware server that caches requests to the WMS map servers boosting the efficiency of the WMS services when used with compatible client applications like those based on OpenLayers or World Wind (by NASA). A Java porting of this application is also integrated into GeoServer.

There are many **proprietary and commercial solutions** that target almost only business market due to high licensing costs: common prizes are, for example, \$10,000 for 2CPUs servers (Webmap) or \$7,500 for 1 CPU plus \$5,000 for each extra CPU (ArcIMS) or \$24,000 for 2 concurrent transactions over unlimited CPUs (GeoMedia). Most known platforms are:

ESRI's **ArcIMS**: claimed to be easy to install and set up and can be used to rapidly build a site with scalable architecture, data integration capabilities and standards-based customization, integration, and communication. It's being dismissed in favour of **ArcGIS for Server**, that integrates with ArcGIS for Desktop which is used to author geographic content. It can also be deployed on Cloud infrastructure and includes APIs in common scripting languages (JavaScript, Flex, Silverlight/WPF) to build and embed interactive maps in websites. Data is delivered in different formats, raster (JPG, PNG, or GIF, no additional client software is required) or vector (Requires a Java plug-in on the client side, using ArcXML). Its advantages are: Out-of-the-box usability, Capability of administering server software from a remote location, integration with other ESRI GIS software. Main disadvantages: needs converters for use non-ESRI data sources and depends on third party software products for being used as middleware.

Autodesk **Infrastructure Map Server** is a web-based mapping software for publishing and sharing CAD, GIS, and other infrastructure asset information via the Internet. It includes templates that enable users to quickly deliver information from AutoCAD Map 3D software to the web. It's the evolution of MapGuide (released as open source) and its main features are: Mobile viewer extension for using Infrastructure Map Server with popular touch-screen devices, GeoREST extension to repurpose existing server and Feature Data Objects (FDO) data using a RESTful web services protocol, WMS & WFS publishing for simplify the publishing process with an OGC WMS & WFS publishing interface, QuickPlot functionality,

Pre-caching of base layer tiles for performance boost and Stylization user interface to create rich cartographic maps.

**MapInfo** developers provided the first tools to Microsoft that allowed them to include mapping functionality in their products and collaborated with Oracle Corporation to develop the original spatial add-on for the Oracle 8i database. The MapInfo client, desktop GIS application, can access data from any standard database supporting the Oracle, Microsoft and IBM software databases. In addition, MapInfo Professional can read and write Oracle Spatial software data types directly. MapInfo SpatialWare extends this same capability to an IBM Informix or MicrosoftSQL server.

Intergraph's **GeoMedia WebMap** delivers data both in raster and vector format: in the second case an additional plug-in must be installed in clients for decoding the ActiveCGM (computer graphics metafile) format. It offers enterprise data access, sophisticated geospatial analysis, and map generation plus powerful linear referencing and analysis capabilities (including routing and dynamic segmentation Web services) and the ability to build a Web application that writes data to Oracle Spatial or Microsoft SQL Server application.

With respect to the desktop GIS applications, also referred to as thick or fat clients, fully functional without a network connection, another type of clients is called **light clients** as they only need a web browser to interact and visualize data produced by Map Servers. This kind of web applications is generally based on Hyper Text Markup Language (HTML) documents with all logics written in client-side scripting language like Javascript or with the implementation of Java applets. These applications can interface with different Map Servers leveraging standard data exchange protocols.

**OpenLayers** (Open Source Geospatial Foundation, e) is an Open Source advanced Javascript library for the creation of light Web-GIS clients developed by MetaCarta. It handles WMS and WFS, has map tiling and proactive cache support and supports, among a wide set of data formats, OpenMaps, Google Maps and Yahoo Maps. According to its developers, OpenLayers is intended to separate map tools from map data so that all the tools can operate on all the data sources. Plug-ins add extra features, like the support for WMS layers with SVG as image format, for SVG enabled browsers. The current release, v. 2.11, have improved support for mobile devices, with a focus on touch interactions, and a general optimization of map rendering performances. The power and versatility of this library is one of the central aspects that make it a very diffuse choice in web GIS development, as in the the case of the implementation hereafter proposed.

**Ka-Map** is a JavaScript library that integrates, server side, with PHP scripts dedicated to the interface with the UMN MapServer. It supports map tiling and caching, and major features are: keyboard navigation options for zooming and panning, zooming to pre-set scales, scalebar, legend and keymap support, query layer data via point and click on the map. Currently the developing seems dismissed, with version 1.0 dated at 2007 and developers version 2.0, with the integration with OpenLayers library, stagnant at 2008.

**Mapbender** (Open Source Geospatial Foundation, c) is implemented in PHP, JavaScript and XML and dual licensed under GNU GPL and Simplified BSD license. It provides a data model and web based interfaces for displaying, navigating and querying OGC compliant map services. It currently supports WMS, WFS(-T) and WMC services. It features KML

support, new interface platforms integrating jQueryUI (jQuery Project, a), WFS improvements, feature encoding, translucency, personalisation, catalogue interface, search module and a compressible directory tree. Current version 2.7 integrates with other Javascript library as OpenLayers (geodata management) and jQueryUI (presentation and server interaction).

**MapFish** (Open Source Geospatial Foundation, d) is based on the Pylons Python web framework; it provides specific tools for creating web services that allows querying and editing geographic objects. It currently offers APIs also for Ruby on Rails and PHP/Symfony via plug-ins. It makes use of OpenLayers for geographic data and of ExtJS (Sencha), the JavaScript component library devoted to user interface design for delivering rich client application experience.

**GeoTools** (Open Source Geospatial Foundation, b) is a Java library for geomatics applications development with functionalities for clients (both light than desktop) and servers. It can manage many data formats (Shapefiles, PostGIS, MySQL, Oracle, ArcSDE, Geomedia, GeoTIFF, ArcGrid and others), and supports many OGC standards (for example WFS, SLD, Filter Encoding). Its main features are: supports OGC Grid Coverage implementation, coordinate reference system and transformation support, symbology using OGC Styled Layer Descriptor (SLD) specification, attribute and spatial filters using OGC Filter Encoding specification, supports Java Topology Suite (see next).

The Java Topology Suite (**JTS**) is a library compliant to the OGC standard Simple Features Specification for SQL and is devoted to 2D topology functions. It offers functions like *union* or *intersection* of shapes and interpretation of topology queries on shape states (e.g. detecting the overlap of 2 or more shapes). Its last release is 1.11 (2010) and presents many portings into other languages: GEOS (C/C++), Net Topology Suite (C#), GeoTools.NET (.NET) and JSTS (JavaScript). PostGIS and GRASS are developed using this library.

## 2.4 Authoring tools

An authoring environment for web GIS provides a tool which is used for the creation of web based GIS applications. This environment/programming method may be based on HTML coding, a scripting language or specific programs created for web GIS applications. All types of authoring environments employ data sources (the spectrum of format depends on the support of file formats and data file standards) and result in various application formats, which either follow an open standard or a proprietary application format.

A very expectable powerful authoring language for web GIS is HTML (Hyper Text Markup Language). Its technical implementation, recommended by the W3C (WWW Consortium) lacks of specific aspects related to geomatics and the adoption of client side scripting libraries like those mentioned in the previous paragraphs, or the implementation of client-server technologies via java virtual machine or ad hoc plug-ins is needed. The presentation layer logics are served using various technologies or various aspects: CSS (Cascading Style Sheets) for styling the appearance of the application, XML (eXtensible mark-up Language) for configurations and/or data exchange between application layers, SVG (Scalable Vector Graphics) for graphic rendering, etc. The extension of HTML with client-side scripting libraries devoted to the development of user interfaces like the mentioned jQuery or ExtJs boosts the interaction with the final user and helps to make thinner the border between web

application and standalone software. Server-side technologies like PHP, Perl, Python or many others are self-standing programming language that can be interfaced by client-side scripts to send and request data or invoke calculations and get results.

Many commercial suites offer authoring platforms for the creation and customization of web facilities in the field of geomatics. They are often the collation of different tools, generally desktop applications, for creating the geographical content, edit the web aspect of the site and perfect its online publication. ESRI, for example, with its **ArcGIS**, lets the user author data, maps, globes, and models on the desktop and serve them out for use on different platforms (desktop, web browser, mobile devices). It also features a set of tools for building custom applications. Bentley offers its **Geo Web Publisher V8i** (Bentley Systems Inc., b) to author and deploy web GIS applications, incorporating drawings, maps, models, aerial photography, and images within custom browser presentations: it can be defined as a desktop GIS editor with an embedded web authoring tool.

**MapXtreme** (Pitney Bowes Software Inc.) from Pitney Bowes is a software development kit (SDK) based on .NET (so MS Windows only) for integrating location intelligence with existing applications and is addressed mainly to business systems. It allows developers to build custom mapping applications and provide tailored views of geographic data. Also Intergraph, with the already cited **Geomedia** Suite, and Cadcorp's **GeognoSIS** (CadCorp) present some tools for the web publication of their map server output.

Also after a detailed review of the existing solutions among free and open source software as well as commercial and proprietary solutions, it seems that a fully web based authoring tool for web GIS applications, that is an authoring application completely accessible and usable simply with an internet connection for the creation of a fully customizable web GIS, seems to be an approach not yet implemented and worth to be further investigated.

	Desktop GIS	WebGIS	Scripting languages	Authoring tools	Our proposal
GeoData modification capabilities	yes	N/A	yes	some	yes
WebGIS creation/exporting capabilities	some systems	N/A	yes	yes	yes
Ease of use	medium	high	low	high	high
Extendability	some systems	no	yes	some systems	yes
Interoperability	high	N/A	high	high	not needed
Availability	low	high	N/A	low	high

Table. 1. A comparison table between the state of the art and the proposed approach to WebGIS generation.

Table 1 summarizes a comparison between our approach and the macro-areas present in the state of the art; the parameters evaluated are:



- GeoData modification capabilities: does the evaluated tool allow its users to modify and update the data used to build the final WebGIS platform?
- WebGIS creation/exporting capabilities: does the evaluated tool allow the creation of a WebGIS platform?
- Ease of use: the degree of usability of the system by users that are not computer scientists
- Extendability: can the evaluated tool be extended with additional features?
- Interoperability: is the evaluated tool easy to be integrated into a workflow comprising other tools for the WebGIS creation?
- It is worth noting that being our proposal a fully integrated environment, spanning from WebGIS meta-design to its use, we consider its interoperability to be of high degree.
- Availability: can the evaluated tool be accessed from different devices or remotely?

### 3. The software shaping workshop methodology for WebGIS platform generation

The Software Shaping Workshop (SSW) methodology, in its general form (Costabile et al., 2006, 2007a; Fogli et al., 2005, Nielsen, 1993) allows to design and develop virtual environments that a) support the activities of users acting a specific role in their community and having a specific application domain; b) are tailorable, customizable and adaptive to the working context; c) support the exchange of information among users belonging to different communities; d) are multimodal and interactive. The methodology is evolutionary and participatory: the final user can customize and evolve his/her own virtual environment and he/she is involved in each step of the system development. The star life cycle model of the product (Hartson & Hix, 1989) is referred, that covers the entire life of the product: each prototype must be evaluated before its development.

The SSW methodology considers the development of two different kinds of workshops: the application workshop and the system workshop. The application workshop is a virtual environment customized to each member of the community, according to his/her performing task and role, to his/her abilities (physical and cognitive) and capabilities, and to his/her culture and language. The system workshop is a virtual environment that permits to customize the application workshop to users' preferences, characteristics and needs. As defined in (Costabile et al., 2007b), we consider meta-design as “a design paradigm that includes final users as active members of the design team and provides all the stakeholders in the team with suitable languages and tools to foster their personal and common reasoning about the development of interactive software systems that support final users' work” (Costabile et al., 2007b).

With this idea in mind, workshops are organized into a three level network in which each member of the design team (software engineers, HCI experts and domain experts) collaborate to design and develop virtual environments customized and tailored for their activity domain and performing tasks:

- at the top, meta-design level, software engineers use a system workshop to create other system workshops in order to permit other software engineers, HCI experts



and domain experts to collaborate to design and development of application workshops;

- at the middle, design level, designers collaborate, using their own system workshops, for designing and implementing application workshops;
- at the bottom, use level, domain experts tailor and use application workshops in order to perform their task.

Each expert is a stakeholder that evaluates the system considering it from his/her perspective biased by his/her different cultural backgrounds, experiences and standpoints of problems. Thus, a communication gap arises among the component of the design team: software engineers, HCI and domain experts adopt different approaches to abstraction and follow different reasoning strategies to model, perform and document the tasks to be carried out in a given application domain; furthermore, each expert expresses and describes such tasks adopting his/her own language and jargon (Fogli et al., 2007; Fischer, 2000).

Communication among the application and the system workshops is supported by an annotation tool. Application workshops' users can, in fact, annotate interface elements to point out to the design team problems or functionalities enhancements they need. At the use level, final users exchange data related to their current task in order to cooperate to achieve a common goal. At the design level, HCI experts and domain experts exchange programs specifying the workshops they are going to develop. HCI and domain experts also communicate with software engineers when it is necessary to develop new tools for supporting their activities. The lower levels are connected to the upper ones by communication paths, allowing final users and designers to interact with other workshops annotating their problems and communicating them to all the experts working in the same SSW network (Costabile et al., 2007a). The SSW methodology allows to design virtual environments in analogy with artisan workshops, i.e. small working environment where artisans such as blacksmiths and joiners manipulate raw materials in order to manufacture their artefacts. Artisans can adapt the environment to their needs by making available all and only the tools they need in the different specific situations. By analogy, the methodology permits to design virtual environments as virtual workshops allowing the user to access sets of virtual tools having a familiar shape and behaviour. Such workshops consent users to perform their tasks and to adapt their virtual working environment using a high-level visual language, manipulating objects in a realistic manner. Final users may act a dual role: the role of consumers when they use the tools offered by the system and they match their needs or they may act the role of designers when they need to adapt the tools to their necessities. Two personalization activities have been recognized in (Costabile et al., 2005): customization, which is carried out by the design team generating application workshops for a specific users community and tailorization, which is the activity performed by the final users adapting an application workshop to particular activities and work contexts. Each actor involved at meta-design level will use a workshop that lets him acquire data from different sources, manage and store it in a repository shared between all the software environments.

Digitization of paper-based cartographic material and data entry tasks will be speeded-up and the results optimized by functionalities such as automatic colours and levels balancing, and auto-completion of recurring data based on already stored information. Another important aspect that has to be taken into consideration at meta-design level is the creation

of metadata: it will be possible for actors involved to compile sets of metadata and associate them to the data, for example pictures or maps.

Metadata and annotations will be implemented according to the W3C Annotea project (W3C, 2001), based on RDF (W3C, 2004) and XPointer (W3C, 2002) standards: RDF is used to specify metadata and annotations, while XPointer is used to locate annotations and the corresponding annotated documents, all of the stored in the shared repository. Content managers and publishers, acting at design level, will exploit their workshops to further manipulate all the information produced at meta-design level: two distinct environments constituting the workshop used at design level can be identified: the *Data Manipulation Environment* (DME), and the *WebGIS Composer Environment* (WGCE). In the DME, content managers can browse the shared repository and retrieve data (together with metadata and annotations) needed for the development of a specific WebGIS instance, that will be exploited by final users at use level. Automatic categorization of documents on the basis of metadata will ease the searching and aggregation process, supported by graphical representations of sets of data, that will be directly manipulated with drag&drop style of interaction. Finally, all the data needed for a specific WebGIS instance will be saved in a dedicated format, creating a “package” ready-to-use to Web publishers. Using the WGCE, Web publishers can interactively build a WebGIS application by graphically compose the layout of the user interface by choosing from a palette HTML elements and their style; these pre-existing elements can be customized and personalized by the Web publishers, to better answer to their needs. The binding between the newly created WebGIS instance’s user interface and the data pertaining to it will be automatically managed by the WGCE, exploiting the package prepared at meta-design level: information such as: points of interest, thumbnail pictures associated to them, searchable elements and other data will be automatically inserted in the proper graphical element of the WebGIS interface.

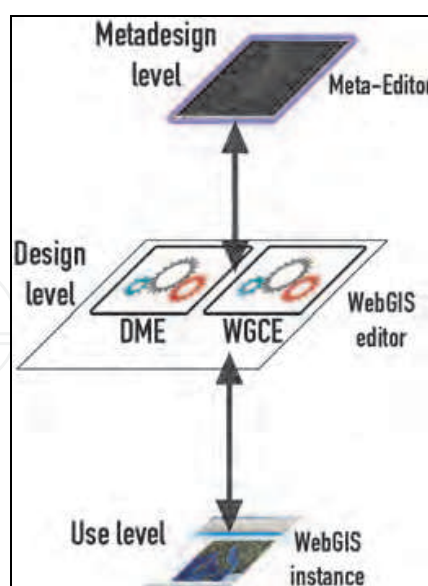


Fig. 1. The SSW methodology specified for WebGIS platform generation.

The Web publisher will then activate the functionalities needed at use level, by accessing a rich set of pre-built computational elements which implement the most popular and used WebGIS-oriented functionalities such as: searching points of interest by specifying

particular parameters (e.g. address, typology, name), compute the route from a point of interest to another, and choosing different map representations. The pre-built set of functionalities will be developed using the OpenLayers Javascript API and exploiting the Web services published by Google: this hybrid approach will permit to add flexibility and extensibility to the core set of computational elements. At use level, final users will interact with the WebGIS instance implemented at design level, exploiting the information and the documents prepared at meta-design level. The adoption of the emerging HTML5 standard, and the use of jQuery-UI library for advanced interaction and animation support will allow to give final users a powerful yet easy to use environment.

4. A System’s architecture proposal

Taking as a starting point the SSW methodology specialized to WebGIS platforms generation, we then propose an architecture encompassing the three levels depicted in Fig. 1. First of all, the main actors of the system and the workflow comprising all the activities they perform are identified and described. The architecture of the network of systems is presented with the help of a UML class diagram showing the main elements of the system and how they interact one with the other. Use case diagrams are then used to more clarify the roles of the various actors and the activities they are involved in.

4.1 Environment heritage conservation and tourism promotion workflow

Let us identify in some more details the four different activities constituting the work cycle (Figure 2): in the first phase, information gathering, the public local administration departments retrieve heterogeneous data about the territory and the services offered at different levels to inhabitants and tourists, their spatial distribution, the viability and all the possible information of touristic nature. Often all the data gathered by the public administration is paper-based cartographic material and needs to be converted in digital form to be subsequently manipulated by exploiting graphic editing tools and, in the case of geographic data manually aggregated using spreadsheet software. The operators of the public administrations (PA operators) are the main actors during this activity.

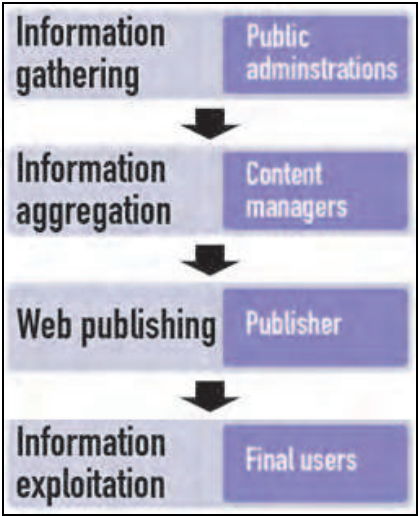


Fig. 2. The workflow for heritage conservation and tourism promotion WebGIS platforms’ development.

In the second phase, information aggregation, data coming from the previous activity is retrieved and classified also on the basis of metadata attached to it by PA operators: different data sources are merged to produce a multi-layered map that comprises all the information regarding a specific resort. Content managers are acting during this phase of the work cycle; they face problems dealing with decoding and working on different data sources using various software tools not inter-operable, thus hindering the cooperation among them.

The structured information encompassed in the multilayered map is passed over the Web publishing phase, in which an environment for building spatially-enabled Internet applications is properly configured to let final users to access the maps through an interactive Web application. Web publishers, the actors involved in these activities, need to configure the Web mapping software application for proper publication of the multi-layered map built by the content managers.

This customization procedure is usually not supported by interfaces designed using a WIMP (Window, Icon, Menu and Pointing Device) interaction style: Web publishers are then forced to carry out activities not properly related to their professional skills such as editing configuration files. The last phase is information exploitation where the final users, which are the stakeholders involved, can browse the multi-layered map in order to gather specific information of their interest, characterized by a high degree of multimediality, expressing complex queries through a graphical interface.

Since Web applications have an international target, their design and development should take into consideration that they will be accessed by users with different cultural backgrounds and speaking different languages. The creation of applications that can effectively be used by an international community is called “internationalization” and is defined in (Dunne, 2006) as “the process of generalizing a product so that it can handle multiple languages and cultural conventions without the need for redesign”. In the process of developing internationally usable software, the activity following the internationalization is called localization, defined in (Dunne, 2006) as “the translation and adaptation of a software or Web product, which includes the software application itself and all related product documentation”.

The architecture we propose describes a system supporting all the stakeholders previously identified by giving them access to custom environments in which they interact with tools they are accustomed to, with a localized interface and offering functionalities for collaborative work.

## 4.2 The proposed architecture

The UML class diagram depicted in Fig. 3 represents the architecture of the network of workshops previously shown in Fig. 1; it specifies the relations between the different software components and how they relies on each other.

The fulcrum is the User Interface class, which is specialized by the three subclasses Meta-Design Environment, Design Environment and Use Environment. User Interface class manages all the aspects related to the interaction between the system and the users: in this case, giving the fact the user interface must support four different categories of users (PA

operators at meta-design level, content managers and publishers at design level, and final users at use level), each one of its subclasses instantiates the different functionalities and tool each environment offers to its users.

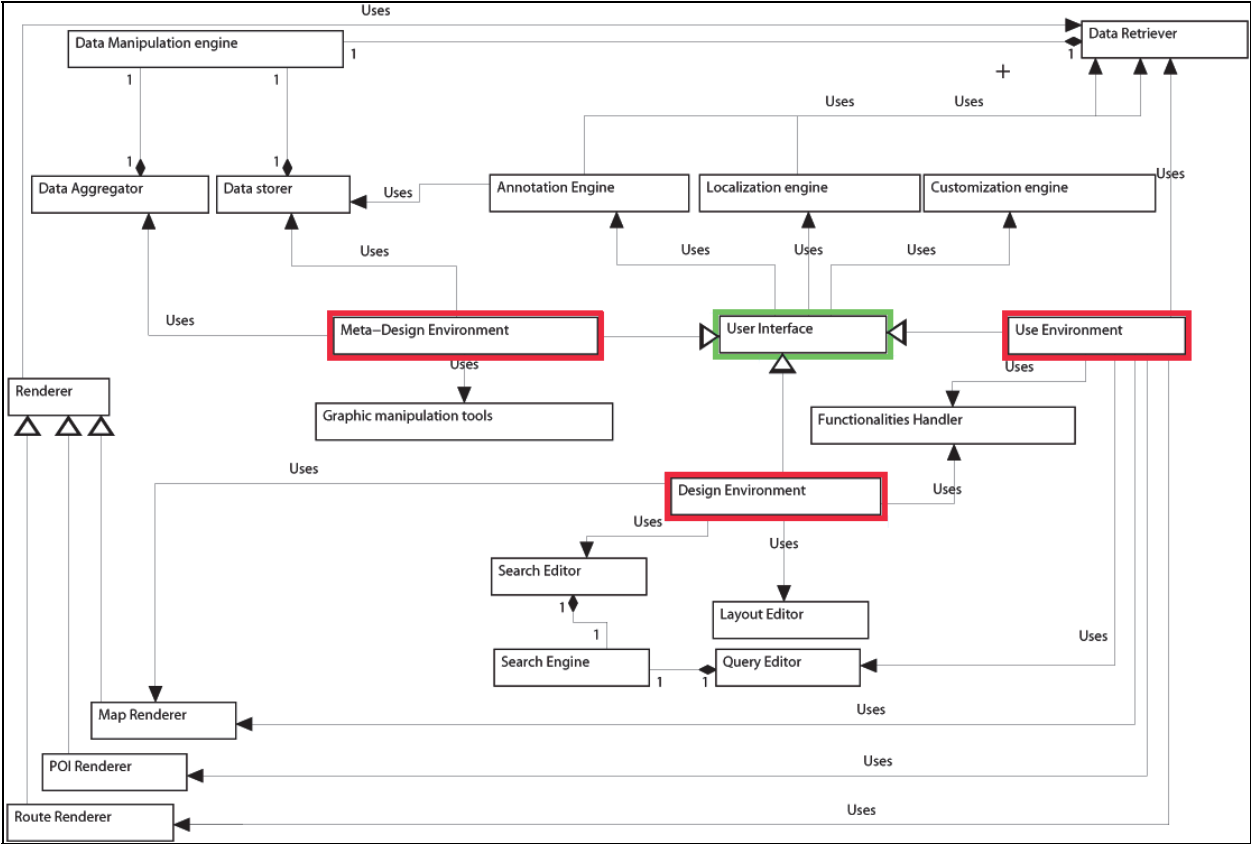


Fig. 3. UML class diagram of the proposed architecture.

All the graphical entities that will be materialized in the interfaces users can interact with, such as icons, buttons, text fields and text labels, are localized according to the preference each user indicates, and customized on the basis of the role of each user, i.e. if s/he is a PA operator, a content manager and so on. The user interface localization process is performed by the Localization engine, which lets users to decide the interface language, chosen from a list of four languages (italian, english, german and french). It relies on XML language files, and can be expanded simply by adding a new language file. Users can choose their language whenever they want by clicking on an icon representing the flag of the language’s country.

In contrast, the customization engine acts automatically on the user interface: contextually to the environment invoking it, it presents widgets and data coherently to the skills and role of the user. For example, PA operators are used to work with spreadsheets, where datasets are represented in a rows-columns fashion: at meta-design level, the environment they interact with recreates this kind of organization, and all the editing operations allowed on datasets, such as add or remove data, organize and annotate data, are offered to the user as operation on rows or columns. At the same time, some of the activities performed by the PA operators involve the use of image manipulation programs: they will then be given a set of tools for image manipulation such as gamma corrections, automatic contrast adjustment and so on: all these functionalities are implemented by the Graphic manipulation tools class.



Content managers, on the other hand, acting at design level will be able to interact with the same data meta-designers have previously gathered and edited, but in terms of geo-referenced points of interest on a map, or trails comprising a subset of the data identified at meta-design level.

Web publishers will then exploit tools to let them graphically organize the representation of the data previously manipulated by content managers in terms of icons to be associated with points of interests, text labels and HTML elements concerning visual organization of information.

The Annotation Engine allows to create and recall annotations produced by the system's stakeholders: annotations are mostly used as a mean of communication between actors of different levels. At design level, content managers can annotate the data produced at meta-design level if they found errors or if they need some kind of information that has not been produced. Doing so, PA operators can retrieve these annotations and correct the problems. At use level, final users can add geo-referenced annotations to the map they are browsing, or read annotations left by other users; this allows them to share their impressions and experiences about places they have visited. Moreover, they can annotate the WebGIS instance itself, to communicate to content managers and Web publishers their needs in terms of information or functionalities available. By retrieving these annotations, actors at design level can improve the WebGIS instance, following the natural co-evolution of users and system.

The Data Manipulation Engine class implements, through the Data Aggregator, Data Storer and Data Retriever, all the functionalities needed by all the actors involved in the workflow to read, edit and write data according to their role in the system: the Data aggregator is used to manipulate datasets and create new ones; this is particularly useful to homogenize data gathered from different sources and create new ones, that will be saved in a suitable format by exploiting the Data Storer. The Data Retriever manages all the operations to access data already stored.

The Renderer class is specialized by three different subclasses, each one of them taking care of rendering, mostly in the WebGIS instance at use level, different geo-referenced graphical elements: the map, the points of interest and the routes.

The Renderer relies on the Data Retriever to get geo-referenced data and annotations to display in the WebGIS instance, and on the Data Storer to save annotations and temporary routes computed by final users.

The Functionalities Handler is a class that allows web publishers at design level to activate and configure the functionalities that will be offered to final users at use level, through the WebGIS instance. They comprise: map zoom and pan, selection of points of interest, allow or deny annotations' creation, route computation and so on.

At the same time, the Functionalities Handler is called at use level to actually implement the selected functionalities.

The Layout Editor acts as a graphical editor for the disposition of the functionalities and the information visualized by final users interacting with the WebGIS instance: content managers can build the WebGIS instance's layout by dragging and dropping widgets associated to

map's browsing tools and visual elements such as page header, page layout and how final users' requested information should be displayed.

Finally the Search Engine, constituted by the Search Editor and the Query Editor, allows content managers to configure the search tool used by final users at use level.

By exploiting the Search Editor, content managers can configure which geo-referenced data can be searched by final users and the way they are allowed to build their queries. Final users can then use the Query Editor present in the WebGis instance to search the desired information.

### 4.3 Meta-design Level

The main actors at this level are the *Public Local Administrators*, pursuing the information gathering, mashing up different information sources provided by diverse providers. They act as the main data providers, retrieving data about the territory and the services offered at different levels to inhabitants and tourists. Data are often heterogeneous in their formats (digital, cartographic, text-based) and in the kind of contents. The Meta-Editor they use lets them manipulate data through tools resembling the ones present in spreadsheet software or image manipulation programs. The result of this operation are one or more data documents that can be saved in various formats such as DBF file format or tab delimited text files, or geo-referenced image files.

Fig. 4 (left) represents the use case diagram for the PA operator actor; as can be seen, the activities s/he can perform are:

- Aggregating data: using the Meta-Editor the PA operator can load different files In Microsoft Excel format, DBF format or tab delimited text format, and manipulate them in order to aggregate them and produce the data files used at design level by content managers
- Accessing annotations: Pa operators can access data annotated by content managers at design level in order to fulfil their requirements
- Adding metadata: metadata can be appended to data files in order to signal particular information about them to content managers, to let them better perform their activities. Metadata is also used to configure DME and WGCE
- Configuring DME and WGCE: by adding metadata at meta-design level, once the data files are sent at design level and loaded into the DME of WGCE, on the basis of the information contained in them, some aspects of these environment are automatically configured. In particular, metadata specify which column of the data files represent longitude and latitude of points of interest, or their classification (i.e. if they represent religious buildings, naturalistic attractions, and so on).

### 4.4 Design level

Two different actors perform their activities at design level: content managers and web publishers.

The *content managers* refine the work performed by PA operators at meta-design level and structure the data in so-called layers, that will be visualized in the WebGIS instance accessed

by final users at use level. They are also responsible to associate icons or text labels to points of interest or routes that will be displayed in the WebGIS instance’s map, often on the basis of some kind of classification of this geo-referenced data.

The *web publishers* set up an environment for displaying spatially-enabled information through the WGCE and provide a customization of the output layout on the basis of the user needs, eventually by mean of a set of tools directly accessible by the stakeholders. In many cases the production cycle needs to be structured in a continuous fashion, being the up to date of the information a main requirement for the global objective. In this scenario a facility for the up-to-date data entry is required in order to achieve a quick response time of the whole data production chain. One or more customized Web interfaces can be set up for the configuration and the upgrading of the system.

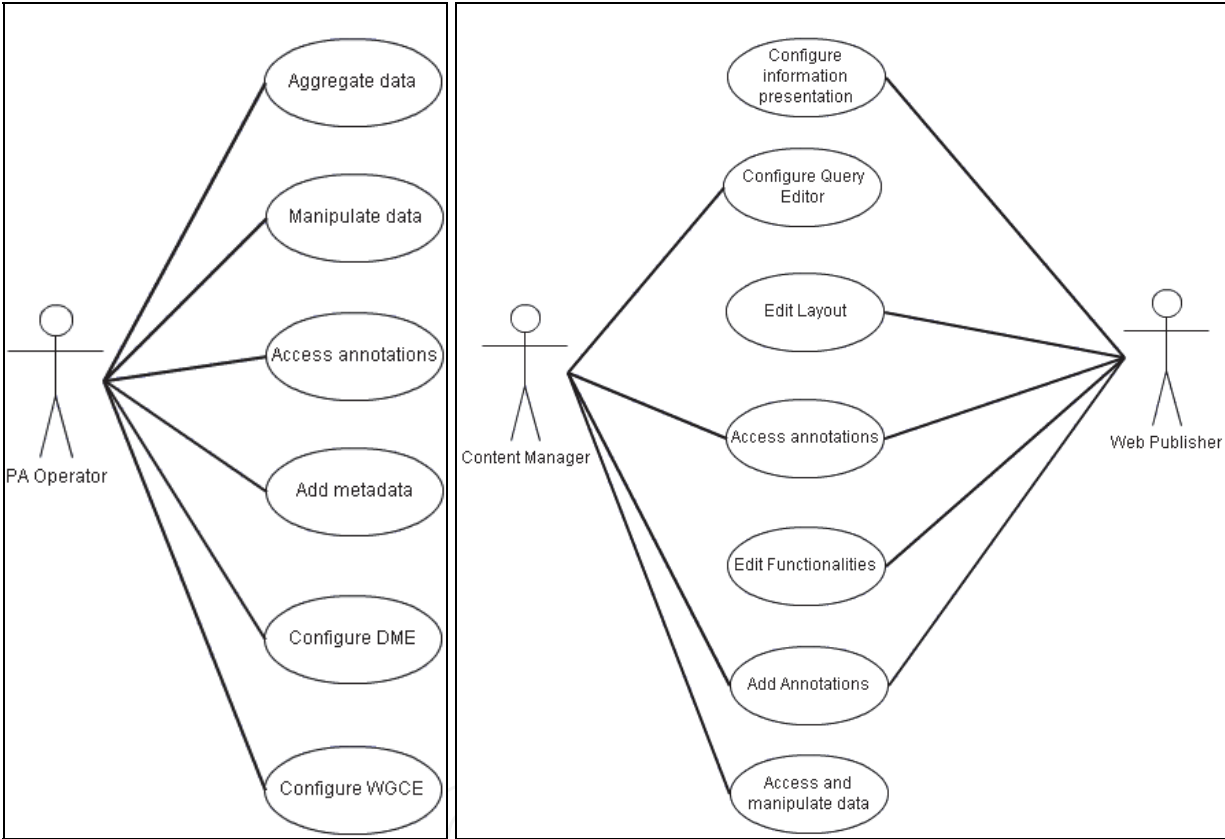


Fig. 4. Left: use case diagram for PA Operator actor. Right: use case diagram for content manager and web publisher actors.

In Fig. 4 (right) is depicted a combined use case diagram for both content managers and web publisher, both of them acting at design level.

Content mangers perform the following activities:

- Accessing and manipulating data: data coming from meta-design level must be accessed by content managers and sometimes needs to be furtherly modified; on the basis of this data they can produce all the layers needed by the WebGIS instance in order to let final users access geo-referenced data under the form of categorized points of interest and routes localizes onto a map.

- Configuring query editor: the query editor will be used by final users to search for specific points of interest or routes: content managers have to configure this functionality by specifying which kind of data can be searched and how: in fact the query editor can be configured to provide final users a search tool based on exact keyword matching, or based on wildcards. Moreover, it can help final users by suggesting possible keywords they might be interested to search for.
- Adding annotations: content managers can add annotations to data produced by PA operators in order to signal to them possible problems or incomplete data.
- Accessing annotations: content managers can read annotation appended to data by PA operators.

Web publisher can:

- Configure information presentation: information that will be displayed in the WebGIS instance must be configured and organized in order to allow an efficient consultation by final users; this is particularly true for information that has to be visualized as a result of a query that has been performed. Web publishers decide where this information will be rendered and its graphical style.
- Edit layout: by exploiting the interactive tools offered by the WGCE, content managers can graphically place into the WebGIS instance the desired functionalities, and can compose its layout by performing drag&drop of elements such as widgets and text labels.
- Adding annotations: just as content managers can do, web publishers can add annotations to data produced by PA operators in order to signal to them possible problems or incomplete data.
- Accessing annotations: content managers can read annotation appended to data by PA operators.

#### 4.5 Use level

The *final users* are the recipients of the information flow and must be made able to gather specific information of their interest, for example browsing a multi-layered map or executing complex queries through a graphical interface, accessing data possibly characterized by a high degree of multimediality.

Once defined the actors involved in the process, a special interest is focused in finding a methodology aimed at designing software environments that allow final users to become designers of their own tools. The proposed approach for achieving this goal is the adoption of SSW methodology (Costabile2007a).

As depicted in Fig. 5, final users perform the following activities:

- search information. By exploiting the search functionality, final users can search the WebGIS instance for specific information by specifying keywords, or selecting keywords from a set of suggested ones. This allow them to locate points of interest or routes on the map that fulfil specific parameters (e.g. searching the WebGIS instance for all the points of interest which fall into the category “churches”);

- visualize geo-referenced information. By browsing the map, or by browsing a list of points of interest or routes, final users can access geo-referenced data such as pictures or descriptions;
- adding annotations. Final users can annotate places they have visited in order to share with other users their experiences, or can annotate the WebGIS instance's interface to communicate to content managers and Web publisher their needs or to suggest enhancements;
- accessing annotations. Final users can access geo-referenced annotations left on the map by other users in order to read impressions and thoughts about places or attractions;
- exploit functionalities. All the functionalities that have been activated by web publishers at design level are available to use by final users; they can be map browsing functionalities or advanced search functionalities (i.e. by using wildcards to specify the keywords to search for);
- compute routes. Final users can compute routes by specifying a starting point and an ending point on the map; the WebGIS instance will automatically elaborate a route that can be the shortest one, or the one comprising more touristic attractions.

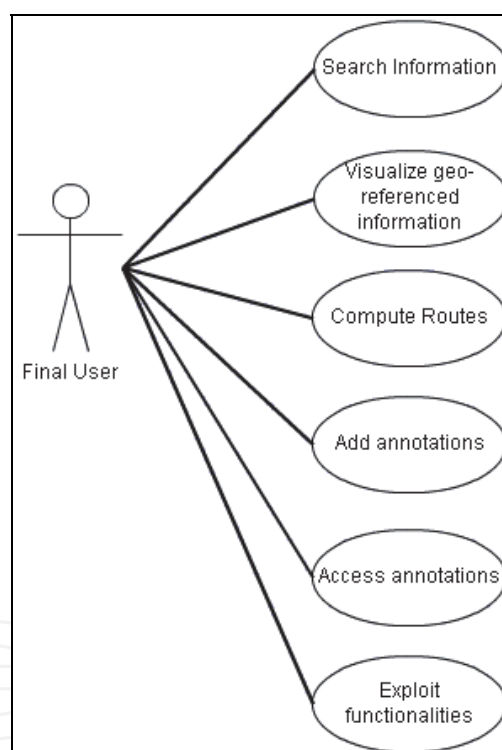


Fig. 5. Use case diagram for final user actor.

## 5. System's prototype description

A prototype of the system based on the architecture described in section 4 has been developed. In its actual development stage, the prototype offers data manipulation functionalities only at design level, while all the meta-design activities are still carried out using an external software. The prototype allows to produce a WebGIS instance for the search and browsing of categorized points of interest localized over a map, and the fruition of multimedia information associated to them.



At design level, content managers can exploit data coming from a dbf file automatically classify points of interest, configure the search functionality offered at use level, specify which icons should represent a particular category of points of interest, specify aliases for particular data fields to be displayed at use level, and indicate the initial state of the map in terms of central latitude and longitude coordinates and zoom factor. Web publishers can act on the overall WebGIS instance presentation by loading a graphical logotype and by inserting a page title and sub-title. Moreover, they configure the information's layout that will be shown to the final users when they select a point of interest. It is a Web-based application supporting collaborative activities among users belonging to the different communities of practice identified in section 3.1, which allow the exchange of documents, generate a shared knowledge base and allow to reach a common goal. The development of the prototype stemmed from the urge manifested by many cities located in the south of Italy to publish a WebGIS application for touristic promotion; they needed a cost effective tool they can easily use to rapidly configure deploy the application, possibly on different hardware platforms and in different places (informative kiosks, local web server, etc.). We propose a technique of interactive systems development based on AJAX (Asynchronous JavaScript and XML), while the OpenLayers API has been used for geospatial data rendering in the Web browser.

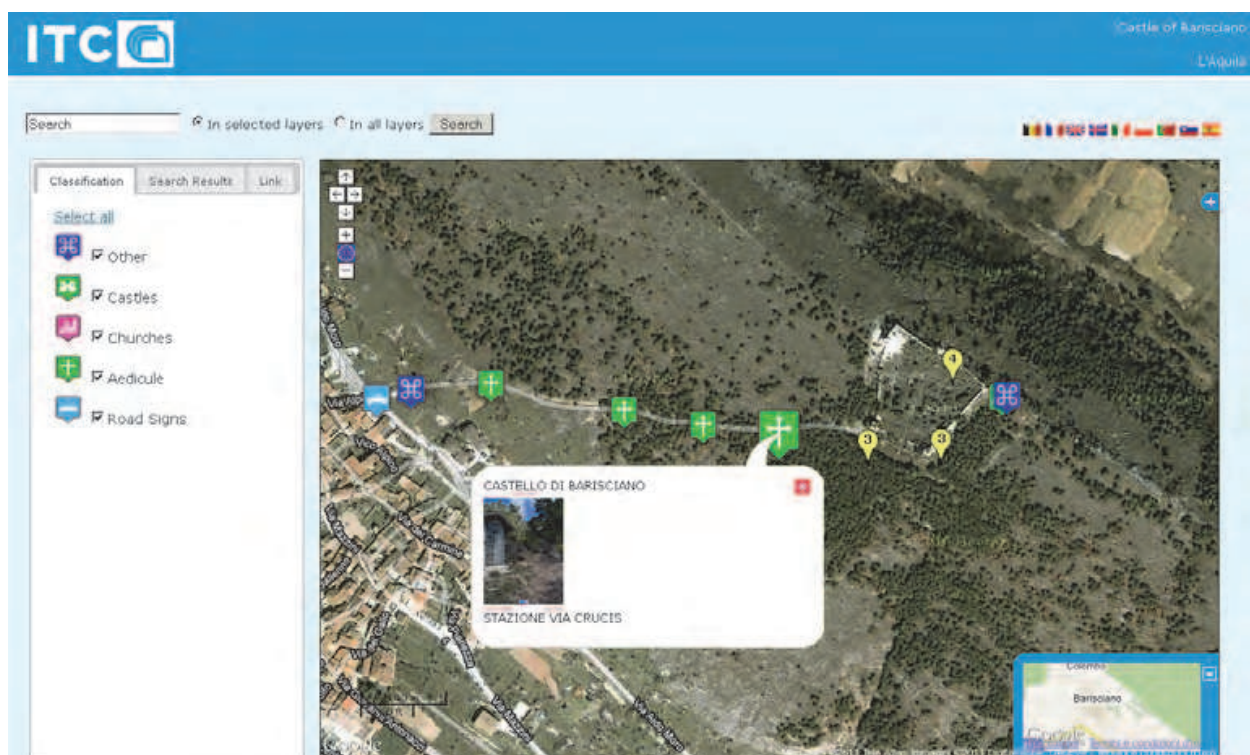


Fig. 6. The resulting WebGIS instance.

The AJAX Web application model involves several technologies: XHTML and CSS for presentation, Document Object Model (DOM) for dynamic display and interaction, XML and XSLT for data interchange and manipulation, XMLHttpRequest for asynchronous data retrieval and JavaScript. To meet the requirement of simplicity for what concerns system's deployment, data is not kept in a database (this would require the installation of a database management system if not present on the Web server) but in XML files, exception made for

geo-referenced data, stored in GeoJSON (Geographic JavaScript Object Notation) format: GeoJSON represents an adaptation of the JSON data-interchange format for expressing geo-referenced data based on a subset of JavaScript language (GeoJSON).

For the realization of an editing environment characterized by a high degree of interaction JQuery UI has been used: built upon the more general JQuery library, it is a JavaScript library providing abstractions for low-level interaction and animation, advanced graphical effects and customizable event handling (jQuery Project, b).

At the same time, JQuery UI has been used for the WebGIS instance user interface (Fig. 7), to provide final users an easy to use set of tools for map browsing and information research.

At this time, the prototype consists of the DME, a limited sets of tool constituting the WGCE, and the resulting WebGIS instance that will be used by final users.

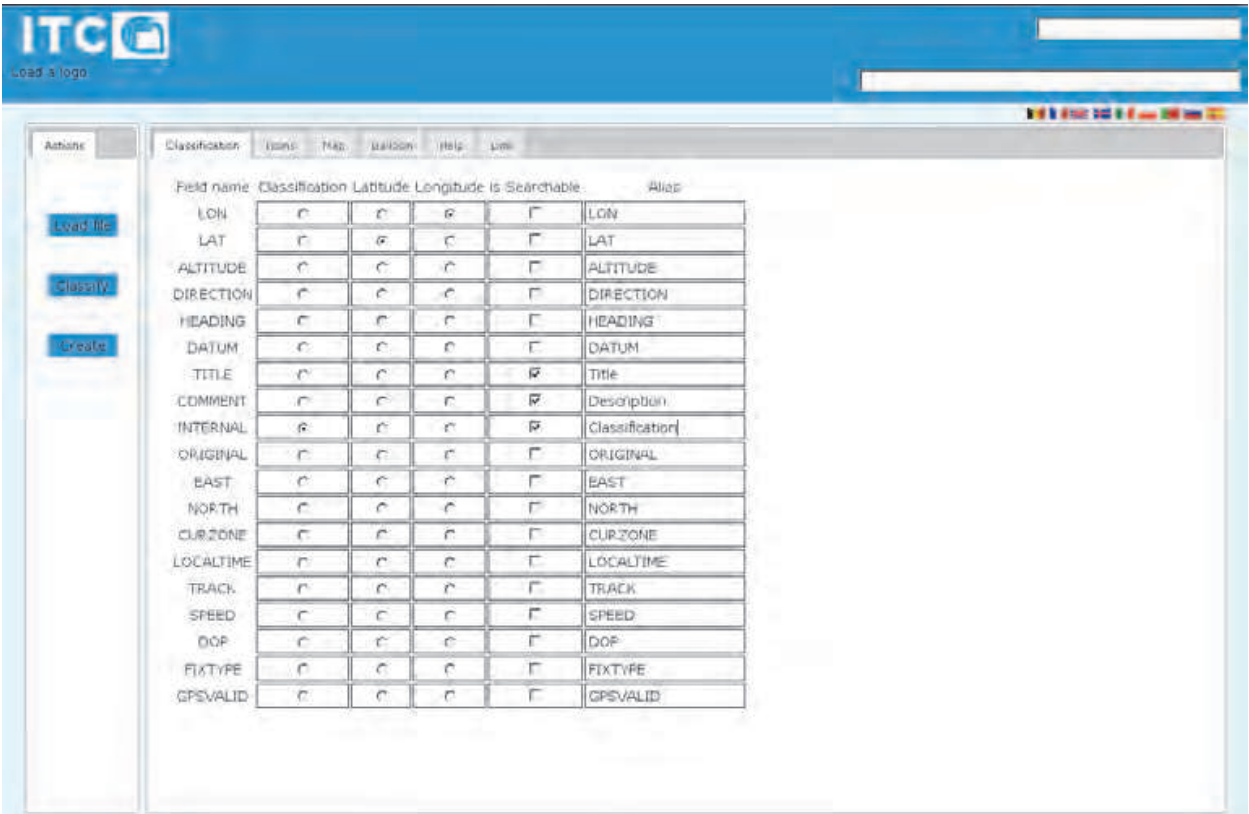


Fig. 7. One of the sub-environments of the DME.

PA Operators send data files in DBF format to the content manager who loads them in the DME by using the “Load file” button located at the far left of the interface shown in Fig. 8. As a result of this loading operation, fields names are visualized as the first column of the central matrix, while the other columns are active, and allow the content manager to define which fields should be used as classification field, longitude and latitude fields, and if a particular field is searchable by final users. Moreover, they can decide to show an alias of the actual field name in the WebGIS instance; this is particularly useful if fields name in the original DBF data file may not be clear to the final user. As an example, in Fig. 8 the content manager have inserted the alias “Description” for the field “COMMENT” which could be confusing.

Once this first activity is finished, the content manager can activate the automatic classification of the geo-referenced data present in the DBF file on the basis of the values found in the chosen classification field and associate to each class of points of interest a meaningful icon and a description by switching to the “Icons” tab (Fig. 8). Every computed class will result in a layer of points of interest of that class in the WebGIS instance.

After the choice of the icons to be associated to each class of points of interest, by switching to the “Map” tab, the content manager can, by acting on a map of the world, decide the starting zoom factor and the center of the map that will be visualized in the WebGIS instance.

Web publishers can, in contrast, decide the layout of the information shown to the final users once a point of interest is clicked on the map: the “Balloon” tab lets them graphically organize the information that will fill a balloon popping up when final users will select a point of interest in the WebGIS instance (as shown in Fig. 8).

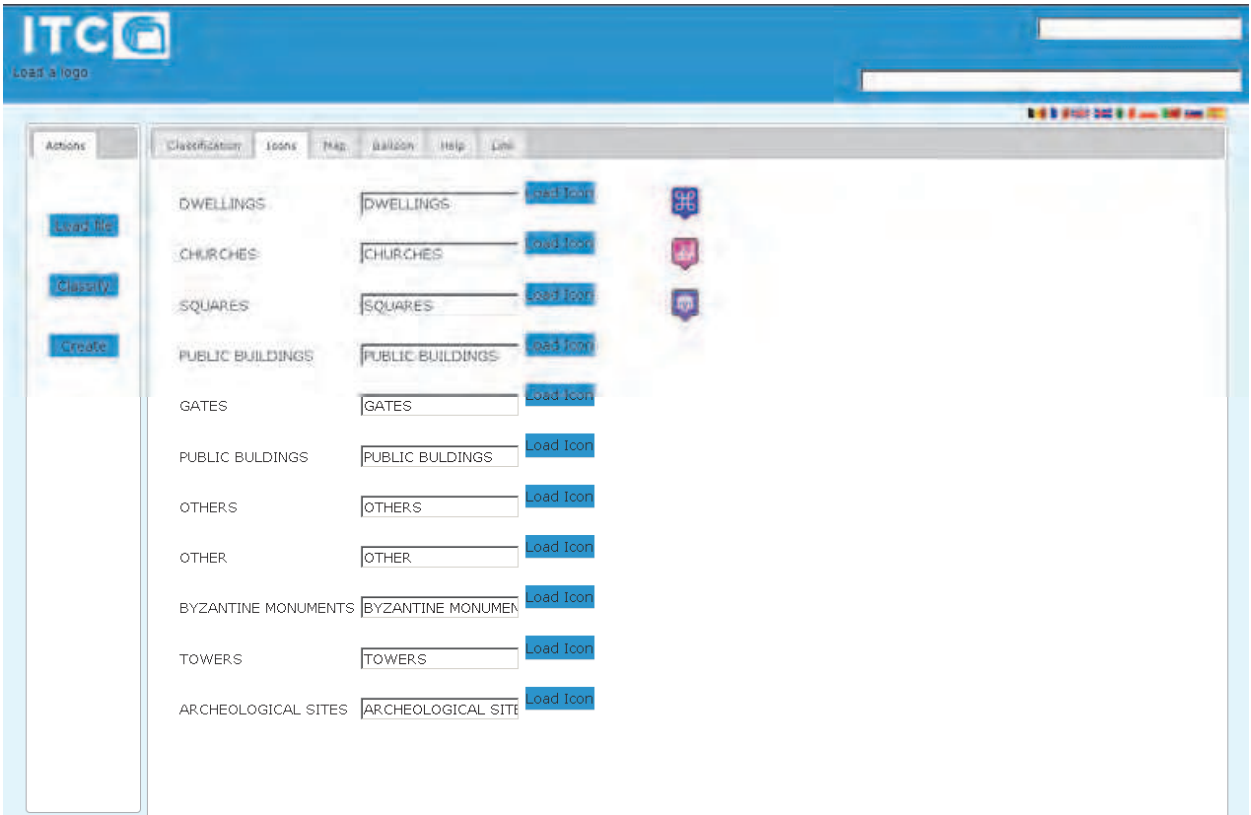


Fig. 8. Icons choice for the automatically computed classification of the points of interest.

Finally, the “Help” tab offers an online help about the functionalities of the system, while the “Link” tab lets web publishers select a set of http links that could be of some interest for final users, and they will be accessible by them by clicking on the “Link” tab in the WebGIS instance.

It is also possible for web publishers to load a logo and specify a title and a subtitle that will be displayed in the heading of the WebGIS instance, by, in turn, clicking on the “Load a logo” text in the upper left and filling the text fields in the upper right.

Once everything has been properly set up, by clicking on the “Create” button in the left column of the interface, the WebGIS instance is created under the form of a ZIP archive file: decompressing it in a Web Server directory will result in the desired WebGIS instance.

Content managers, web publishers and final users can also localize the interface they interact with by clicking the flags located in the upper right of the screen; for the prototype have been produced localization files for English and Italian language.

For what concerns the WebGIS instance, final users can browse the central map and click on points of interest to display the geo-referenced information attached to them, or they can search for specific points of interest by using the search field in the upper left of the screen: they can choose to search for particular keywords in all the classes’ layers or just in the selected ones by acting on the radio buttons located between the search field and the search button. By deselecting a layer (clicking on the checkbox associated to it in the left column), points of interest belonging to that layer will be hidden.

The results of the search activity will be displayed in the “Search Results” tab.

The adopted approach has some benefits compared to the existing solutions for this kind of applications (see Table 1). First of all the designed prototype offers a single development environment for each authoring phase of the project. Most of other authoring tools for Geographic applications, for example Geo Web Publisher by Bentley Systems, are desktop GIS applications for editing the map data and an embedded web editor for the customization of the presentation layer. In other cases the authoring workflow must be split and edited using several different applications of the software suite, like in the Esri suite or in MapXtreme by Pitney Bowes or again in the Geomedia Suite. It must be pointed out that all surveyed solutions use a hybrid approach consisting of a desktop application for map data editing and often (but not always) a web application for web interface authoring: a single environment completely accessible via web is a novelty and a useful approach under many points of view. In the first instance the interaction with a single user interface is more user friendly than forcing the user to make different authoring steps with different applications. A web application, moreover, is, by definition, accessible from anywhere and with almost any hardware capable of running a modern web browser: this makes the proposed solution a versatile approach for on-the-run authoring needs or in case of low computational power devices.

## 6. Conclusion

This chapter proposes a novel approach to the development of WebGIS platforms for tourism promotion and heritage conservation. By adopting a point of view that takes into account not only information fruition by final users, but considering the whole workflow comprising activities such as data harvesting and manipulation, their management and publishing, it is possible to provide a network of integrated environments supporting all the stakeholders involved in the process with dedicated tools. A broad state of the art underlines the large availability of technologies that can be exploited to develop this kind of system, and at the same time reveals the lack of proper authoring environments. According to the SSW design methodology a system architecture has been proposed and discussed, and the actors of the system have been described in terms of how they interact with the



system. Finally, a prototype of the system offering a subset of the designed functionalities is described.

## 7. Acknowledgment

Giuseppe Fresta, Marco Padula and Paolo L. Scala would like to dedicate this chapter to the memory of their master and friend Piero Mussio: without his precious guide this work couldn't be possible.

## 8. References

- Barricelli, B. R., Maiellaro, N., Padula, M., & Scala, P. L. (2008). A collaborative system for environment and tourism information authoring and Web publishing: an institutional case study, *Proceedings of ICIW 2008*, Athens, Greece, June 2008.
- Bentley Systems Inc. n.d. Bentley Map, 18 July 2011, Available at: <http://www.bentley.com/en-US/Products/Bentley+Map/Product-Overview.htm>
- Bentley Systems Inc. n.d. Bentley's Geo Web Publisher, 18 July 2011, Available at: <http://www.bentley.com/en-US/Products/Bentley+Geo+Web+Publisher/>
- Böhner, j. n.d. SAGA, 18 July 2011, Available at: <http://www.saga-gis.org>
- CadCorp. n.d. GeognoSIS, 18 July 2011, Available at: [http://www.cadcorp.com/products\\_geographical\\_information\\_systems/geogno sis.htm](http://www.cadcorp.com/products_geographical_information_systems/geogno sis.htm)
- Costabile, M. F., Fogli, D., Lanzilotti, R., Marcante, A., Mussio, P., Parasiti Provenza, L., & Piccinno, A. (2007). Meta-design to Face Co-evolution and Communication Gaps Between Users and Designers. *Lecture Notes in Computer Science*, Vol. 4554, pp. (46-55), 9783540732785.
- Costabile, M. F., Fogli, D., Lanzilotti, R., Mussio, P., & Piccinno, A. (2005). Supporting Work Practice through End User Development Environments, Technical Report, Bari, Italy: Università degli Studi di Bari, Dipartimento di Informatica, October 2005.
- Costabile, M. F., Fogli, D., Mussio, P., & Piccinno, A. (2006). End-user development: the software shaping workshop approach. In *End User Development Empowering People to Flexibly Employ Advanced Information and Communication Technology*, H. Lieberman, F. Paternò and V. Wulf (Eds.), pp. 183-205, Dordrecht: Springer, 9781402042201.
- Costabile, M. F., Fogli, D., Mussio, P., & Piccinno, A. (2007). Visual interactive systems for end-user development: A modelbased design methodology. *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, Vol 37, No. 6, (2007), pp (1029-1046).
- Django Software Foundation (2005). GeoDjango, 18 July 2011, Available from: <http://geodjango.org/docs/db-api.html>
- Dunne, Kieran. 2006. *Perspectives on Localization*. American Translators Association, John Benjamins Publishing Co., 9027231893.
- Ellis, C. A., Gibbs, S. J., & Rein, G. L. (1991). Groupware - some issues and experiences, *Communications of the ACM*, Vol. 34, No. 1, pp. (39-58).
- ERDA Suite, 18 July 2011, Available at: <http://www.erdas.com>
- ESRI, 18 July 2011, Available at: <http://www.esri.com/>



- ESRI, n.d. ArcSDE, 18 July 2011, Available at:  
<http://www.esri.com/software/arcgis/arcade/>
- ESRI, n.d. ArcView, 18 July 2011, Available at:  
<http://www.esri.com/software/arcgis/arcview>
- Fischer, G. (2000). Symmetry of ignorance, social creativity, and meta-design. *Knowledge-Based Systems*, Vol. 13, No. 7-8, pp. (527-537), 09507051.
- Fogli, D., Fresta, G., Marcante, A., Mussio, P., & Padula, M. (2005). Annotation in cooperative work: from paper-based to the web one, *Proceedings of International workshop on annotation for collaboration*, La Sorbonne, Paris, France, November 2005.
- Fogli, D., Marcante, A., Mussio, P., & Parasiliti Provenza, L. (2007). Design of visual interactive systems: a multi-faced methodology, *Proceedings of Converging on a Science of Design through the Synthesis of Design Methodologies*, San Jose, CA, USA, May 2007.
- GDAL (2011). GDAL Data Model, 18 July 2011, Available at:  
[http://www.gdal.org/gdal\\_datamodel.html](http://www.gdal.org/gdal_datamodel.html)
- GDAL (2011). OGR, 18 July 2011, Available at: <http://www.gdal.org/ogr/>
- GDAL, n.d. GeoJSON GDAL, 18 July 2011, Available at:  
[http://gdal.org/ogr/drv\\_geojson.html](http://gdal.org/ogr/drv_geojson.html)
- GeoJSON, 18 July 2011, Available at: <http://geojson.org/geojson-spec.html>
- GeoMedia, n.d. 18 July 2011, Available at:  
<http://www.intergraph.com/cgi/products/productFamily.aspx?family=10&country=>>
- GeoServer (2008). 18 July 2011, Available from:  
<http://svn.codehaus.org/geoserver/tags/2.0.0-alpha1/geoserver/release/README.txt>
- Groot, R. & McLaughlin, J. (2000). *Geospatial Data Infrastructure, Concepts, Cases and Good Practice*, Oxford University Press.
- Hartson, H. R., & Hix, D. (1989). Human-computer interface development: concepts and systems for its management. *ACM Computing Surveys (CSUR)*, Vol. 21, No. 1, pp. (5-92).
- Institut de Recherche Sciences et Techniques de la Ville (IRSTV). n.d. OrbiGIS, 18 July 2011, Available at: <http://www.orbisgis.org>
- jQuery Project. n.d. jQuery, 18 July 2011, Available at: <http://jquery.com/>
- jQuery Project. n.d. JQuery UI, 18 July 2011, Available at: <http://jqueryui.com>
- Jump Pilot Project, n.d. OpenJUMP, 18 July 2011, Available at:  
<http://www.openjump.org/>
- Konecny, G. (2003). *Geoinformation: Remote Sensing, Photogrammetry and Geographic Information Systems*. (1st ed.) Taylor & Francis, London
- Longley, P.A. (2001). *Geographic Information Systems and Science*, (1st ed.), John Wiley & Sons, Chichester
- MapWindow Open Source Team, n.d. MapWindow, 18 July 2011, Available at:  
<http://www.mapwindow.org>
- MetaCarta, n.d., TileCache, 18 July 2011, Available at: <http://tilecache.org>
- Nielsen, J. (1993). *Usability Engineering*. Academic Press, 0125184069, San Diego.

- Open Geospatial Consortium. n.d. Simple Feature Access - Part 2: SQL Option., 18 July 2011, Available at: <<http://www.opengeospatial.org/standards/sfs>>
- Open Geospatial Consortium (2005). WFS Web Feature Service Implementation Specification, 18 July 2011, Available At: <[http://portal.opengeospatial.org/files/?artifact\\_id=8339](http://portal.opengeospatial.org/files/?artifact_id=8339)>
- Open Geospatial Consortium (2006) GeoRSS White Paper, 18 July 2011, Available at: <<http://www.opengeospatial.org/pt/06-050r3>>
- Open Geospatial Consortium. (2006). OpenGIS Geography Markup Language (GML) Implementation Specification, 18 July 2011, Available at: <[http://portal.opengeospatial.org/files/?artifact\\_id=20509](http://portal.opengeospatial.org/files/?artifact_id=20509)>
- Open Source Geospatial Foundation (2006) Tiling standard, 18 July 2011, Available at: <<http://wiki.osgeo.org/index.php/TilingStandard>>
- Open Source Geospatial Foundation, nd. degree map server, 18 July 2011, Available at: <<http://www.deegree.org/>>
- Open Source Geospatial Foundation, nd. GeoTools, 18 July 2011, Available at: <http://www.osgeo.org/geotools>
- Open Source Geospatial Foundation, nd. Mapbender, 18 July 2011, Available at: <<http://www.mapbender.org>>
- Open Source Geospatial Foundation, nd. MapFish, 18 July 2011, Available at: <<http://mapfish.org/>>
- Open Source Geospatial Foundation, nd. OpenLayers, 18 July 2011, Available at: <<http://openlayers.org/>>
- Open Source Geospatial Foundation, nd. OSSIM, 18 July 2011, Available at: <<http://www.ossim.org>>
- Open Source Geospatial Foundation, nd. QuantumGIS, 18 July 2011, Available at: <<http://www.qgis.org/>>
- Oracle, n.d. Oracle 11g Spatial 18 July 2011, Available at: <<http://www.oracle.com/technetwork/database/options/spatial>>
- Pavlenko, A. (2009). Mapnik python module, 18 July 2011, Available at: <[http://svn.mapnik.org/tags/release-0.6.0/docs/api\\_docs/python/mapnik-module.html](http://svn.mapnik.org/tags/release-0.6.0/docs/api_docs/python/mapnik-module.html)>
- Pitney Bowes Software Inc. n.d. MapExtreme 2008, 18 July 2011, Available at: <http://www.pbinsight.com/products/location-intelligence/developer-tools/desktop-mobile-and-internet-offering/mapxtreme-2008/>
- PostgreSQL Global Development Group. n.d. PostgreSQL, 18 July 2011, Available at: <<http://www.postgresql.org/>>
- Refraction Research. n.d. uDig, 18 July 2011, Available at: <<http://udig.refrations.net/>>
- Safe Software. n.d. 18 July 2011, Available at: <[http://www.safe.com/reader\\_writerPDF/geojson.pdf](http://www.safe.com/reader_writerPDF/geojson.pdf)>
- Sencha. n.d. ExtJs, 18 July 2011, Available at: <http://www.sencha.com/products/extjs/>
- SUN Microsystems. n.d. MySQL, 18 July 2011, Available at: <http://www.mysql.com/>
- SUN Microsystems. n.d. MySQL Spatial Extensions , 18 July 2011, Available at: <<http://dev.mysql.com/doc/refman/5.5/en/spatial-extensions.html>>
- University of Minnesota, n.d. MapServer, 18 July 2011, Available at: <<http://mapserver.org/>>

- W3C. (2001). Annotea, 18 July 2011, Available at: <http://www.w3.org/2001/Annotea/> Last accessed 2011-03-01
- W3C. (2002). XPointer, 18 July 2011, Available at: <XML Pointer Language, <http://www.w3.org/TR/xptr/>>
- W3C. (2004). RDF, 18 July 2011, Available at: <Resource Description Framework, <http://www.w3.org/RDF/>>

IntechOpen

IntechOpen



## **Human Machine Interaction - Getting Closer**

Edited by Mr Inaki Maurtua

ISBN 978-953-307-890-8

Hard cover, 260 pages

**Publisher** InTech

**Published online** 25, January, 2012

**Published in print edition** January, 2012

In this book, the reader will find a set of papers divided into two sections. The first section presents different proposals focused on the human-machine interaction development process. The second section is devoted to different aspects of interaction, with a special emphasis on the physical interaction.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Davide Di Pasquale, Giuseppe Fresta, Nicola Maiellaro, Marco Padula and Paolo Luigi Scala (2012). New Frontiers for WebGIS Platforms Generation, Human Machine Interaction - Getting Closer, Mr Inaki Maurtua (Ed.), ISBN: 978-953-307-890-8, InTech, Available from: <http://www.intechopen.com/books/human-machine-interaction-getting-closer/a-distributed-system-for-webgis-platforms-generation>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](https://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen