Rapid GIS Development: a model-based approach focused on interoperability

Rui Cavaco, Rui Sequeira, Mário Araújo, Miguel Calejo
SIG2000, Declarativa

ABSTRACT

We present a framework for rapid development of GIS web applications tightly integrated with web database applications, with several innovations: a georeference abstraction, simplifying use of complex geographical geometries; use of SQL column data types to specify the user interface without scripting; interface and symbology personalization, persisting in user preferences; and a flexible implementation based on open source components, including full auditing. The framework is already used in a mission critical application.

CONTEXT

Rapid development frameworks for GIS were slowly coming to life, while developers waited for GIS software market leaders' moves. Currently, *open source* initiatives like University of Minnesota's MapServer, OpenLayers graphic web client library and others have shed a whole new light that drove the creation of plenty of small frameworks with limited objectives and which were developed as proprietary tools, not as full-fledged products.

On another area, rapid web application development tools abound, such as our own Web Application Maker[1] ("WAM"), powering web information systems in institutions in Portugal and the USA. This tool provides alphanumerical forms with minimal effort, but not any[2] GIS capabilities. One such web information system, the backbone application of Portugal's Northern Region Commission[3], required incorporation of geographical information associated to its detailed data records, and close integration with GIS editing tools.

NEW APPROACH

The authors joined efforts to bring up a new RAD approach to database-centric GIS applications, based on the following principles:

- Generalize easy to use graphic editing functionalities over a geographic database, removing the need for specialized and expensive GIS editor tools operable only by specialists
- Provide historical record capabilities and full traceability/auditing of graphic editions
- Allow the development of geographical web clients easily interoperable with other web applications, like workflow and document management systems, with an integrated navigation experience

[1] www.declarativa.pt/wam
[2] Other than basic support for GoogleMaps points
[3] Comissão de Coordenação e Desenvolvimento da Região Norte's Expedientíssimo, its transversal document and workflow management SQL Server + web application .

- Specify the basics of GIS/non-GIS web interface integration without coding, simply by using a SQL user data type ('WAMgeoref') in each application table incorporating geographical columns

- Share authentication of existing web applications

The proposed RAD solution implements the concept of *Generic Georeference (*materialized in the abovementioned SQL type*):* a single integer identifier (a table record foreign key value), shareable between applications, identifies a polymorphic georeference, which can aggregate one or more georeferences of different types: point, line, polygon, or *external file*. External file meaning a raster or an ESRI Shapefile format datafile.

The implementation of generic georeference concept is supported in a series of database tables which allow to relate a key (or a key list) provided by a foreign application to either one or a collection of geodatabase features or a georeferenced raster or vector file. This is intended mainly for display purposes. When geoprocessing or analysis show up then this generic georeference must be restricted to geodatabase features.

This is a useful concept when, for example, trying to integrate a GIS client into an existing building permit management application. Related to a proposed building permit we should have in readily available semantic data like the cadastre, the proposed building areas, etc. These can be served as structured geodatabase polygons, or, instead we could see solely a CAD drawing with all these elements expressed visually. In the first case we can clearly make good use of some geoanalysis functions, possibly automated, that could deliver immediately if the permit is admissible.

IMPROVING THE USER WEB GIS EXPERIENCE

Existing web GIS applications and frameworks usually provide a very different user experience from desktop (non-web) user interfaces.

The authors considered important to bring the desktop user experience to web clients, such that clients built with this framework can work like this: at first time access, the user views a standard thematic base content and a full geographic extent encompassing all displayed content. The user is then allowed to add new themes. The user is then able to edit symbol parameters for these themes. Such parameterization, including geographic extent, is automatically added to user profile/preferences information and is shown again in the next user work session.

On our framework, the user:

- Is allowed to customize the thematic menu beyond the simple toggling of each theme on or off, adding or removing themes and customizing symbology.

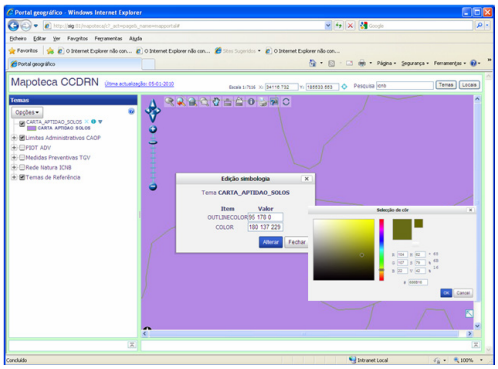

Fig. 1.

- can retrieve viewing settings from a previous work session, retrieving the thematic menu contents, a zoom level and a location.

Adding themes to thematic menus (some times called *table-of-contents)* can become difficult if the supporting geodatabase is loaded with hundreds of GIS themes. In this case, tools were developed to help the user find the needed themes by:

- Keyword searches, which can be based on explicit relations between themes and keywords or by automatic scanning of occurrence of keywords in themes' metadata.

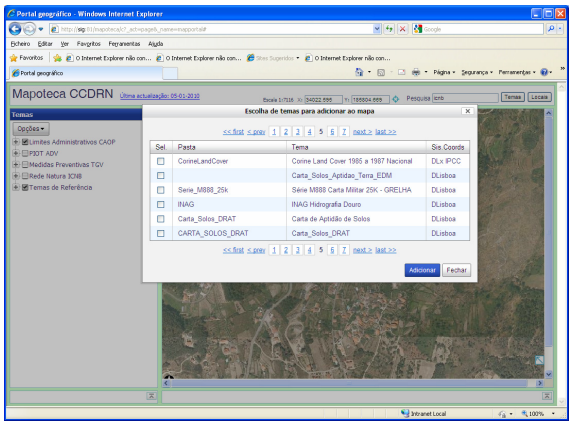

Fig. 2.

- Envelope searches, which test eventual superposition between geodatabase themes and the current map view extent

In our view, all this features are fundamental to really disseminate application-oriented web GIS.

PRINCIPLES IN PRACTICE

It was fundamental in our approach to guarantee the five principles of *interoperability, minimize coding, generalize graphic editing, history/auditing* and *authentication.*

Nowadays saying *interoperability* usually means *webservice invocation*. Webservices are valuable especially when two applications, not sharing a common database, need to exchange structured data with some complexity. This is not our case: when implementing the *generic georeference* concept (see ahead) a GIS web client only needs to share a key value. Webservice invocation in this context would result in unneeded overhead. A third party application can call our GIS client in a separate browser window or on a dedicated IFRAME element simply by invoking a URL and passing the referred key and an URL callback as parameters (or a list of these).

Coding minimizing – the web GIS client was built on OpenLayers and Yahoo User Interface javacript libraries, and was designed to include a broad array of tools and functions some of which are hidden by default and can be simply enabled and parameterized by changing parameter and configuration files in server. Map images are configurable also using MapServer's map file syntax, being now offered a tree-view like editor for easy configuration. Non-standard extensions and graphic tools can still be added through a provided extension mechanism which also contemplates server code extensions in Python language.

On our framework, graphic editing is heavily supported on OpenLayer's library capabilities. Geometries edited with OpenLayers tools are easily exported to GeoJSON and sent to application server in this format. Graphic editing, is technically simple nowadays, but some GIS vendors still impose draconian licensing terms and prices since this used to be available only in top class desktop products. That's not really technically justifiable anymore and, in this framework, the authors actively succeeded to bring an end to this vision. Another unjustified argument used to be that unskilled operators could fill geodatabases with garbage. This is also a possibility with any other database application. Providing auditing and enforcing authentication (see ahead) we are actively combating improper use of applications.

About auditing, the framework allows for use of standard auditing tables. Authentication has been achieved on Windows environments with standard NTLM/Active Directory/"Integrated Windows Authentication". The framework allows for different modules on which the same user can have different permissions and be served with differentiated tools.

APPLICATION

The next figure displays a web form generated by WAM, with a georeference icon (red) providing navigation to the polygon display on the right window; from the GIS window on the right the data record on the left can be open too, so navigation is bidirectional.

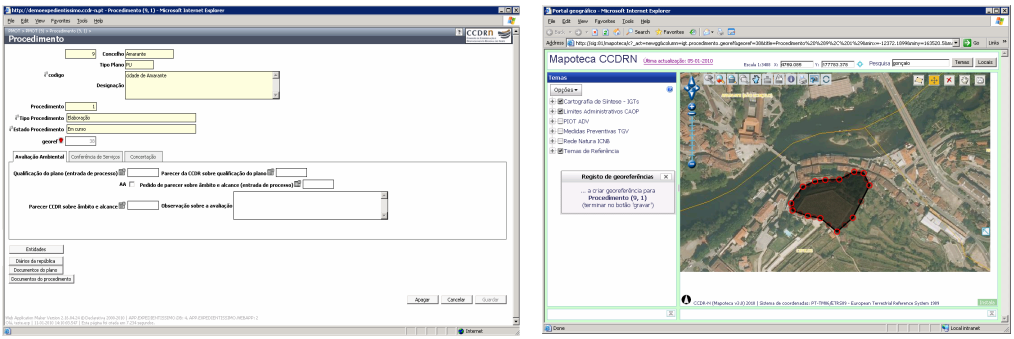

Fig. 3

No coding/scripting was needed: the developer only had to type one SQL column (in the table underlying the left form) as "georeference", and the system made all GIS editing and displaying available to the user.

We're now finishing testing of a similar combination for record sets, allowing a (WAM-based) application list, with arbitrary joins and user specified filter/search conditions, to invoke the GIS window with (only those) georeferences shown; again, with navigation back to application records.

The whole system is being deployed to over 100 in-house users, with likely expansion to extranet users later.

BIBLIOGRAPHY

Butler, H. et al., The GeoJSON Format Specification (http://geojson.org/geojson-spec.html), 2008

Calejo, Araújo and others,, Web Application Maker Manual, http://www.declarativa.pt/wam, 2009