Omega for multilingual mapping

Blanca Mancilla

School of Computer Science and Engineering
The University of New South Wales
UNSW SYDNEY NSW 2052, Australia
mancilla@cse.unsw.edu.au
http://www.cse.unsw.edu.au/mancilla

1 Abstract

This paper presents the design and development of an online geographical mapping tool, producing different versions of a map according to user requirements. Using intensional programming, this mapping server handles the variance in maps, by controlling the flow of map parameters to the component tools. The mapping server currently produces multilingual maps, using Omega as the typesetting facility. Future work includes a more sophisticated user interface, better interaction with Omega, and more sophisticated storing mechanisms for multiple names.

2 The mapping server

The Web-based mapping server combines the typesetting capabilities of Omega [2], the PostgreSQL database management system [1], GMT [5] for creating maps, the lot glued together using ise [4], a context-aware language, which provides a Web interface to the user. The server’s main function is to create a map based on options entered by the user. These options can be mapping parameters (i.e. geographical region, type of boundary, level of detail, color), or language (of the map or of the Web interface, the gui). The values of these parameters define a context and among other things, this context determines the control and data flow for the component applications.

The gui is a multidimensional Web page that changes its presentations according to the context. It is implemented using ise [4] and at present it is a multilingual page — English, Spanish, French — that rediscovers itself when the language changes. Figure 1 shows the page in English, the default language. Pressing on the language interface button shows the available languages of the interface. Figures 2 and 3 show the same page in Spanish and French.

Additional features include adjustment of the interface according to the level of expertise of the user or the level of detail required. ise is a Perl-like scripting language that allows the multiple definition of variables (versioned definitions), which depend on specific contexts. The page is just one file (not three different copies in different languages) and in the indicated places, the value of a variable is evaluated according to the context to give the right content.

GMT [5, 6], a free software package, is the mapping tool used to produce the geographical maps. It has an internal database of world borders in different scales which is used as a default for the mapping server. To plot strings in maps, GMT uses a function called ptext, which is a primitive typesetting mechanism, not able to handle many of the Asian languages. As a result, the author wrote a new function called
psomega, that takes care of all of the typesetting within the map using Omega. This opens up the possibilities to use almost any language.

To manage multilingual names of places we used Postgres, a free database management system (DBMS). Currently we have stored the names of the continents and oceans in seven languages (English, Spanish, French, Tamil, Malayalam, Hindi, Farsi) using Unicode. The tables in the database schema are the location table, languages, and for each language, a names table. When a request is made to retrieve names in a region, the dimension $lgMap$ in the context (language of the map) is evaluated to pick the correct language table and select the names — encoded in Unicode — that fit the request. The output is a series of strings with a (longitude, latitude) pair associated to each string. With this, the mapping tools can calculate the PostScript coordinates (on paper) and typeset the text strings using Omega.

3 Motivation

If you are looking for the geographical location of a place, your main choices are to bring out the thick atlas in the bookcase, to browse on your electronic atlas, or to use the Internet. The first choice will usually be enough if you are content with what the paper shows: a set language, maybe two; information as current as the date of the printed material; a fixed size (i.e. fixed resolution) decided upon before printing. Your electronic atlas could be a bit better: the resolution can change but only to a certain extend. With the Internet, validity of data might seem to be an easy problem to solve, reality is different.

If the user’s enquiries need the map to be rebuilt by changing the projection and language for example, using up-to-date data, our system is the only free one that the author knows of that is capable of dealing with variance in the map parameters, variance of different nature, to produce and render a reliable map. The mapping server displays maps that are created on the spot, not prestored. This fact adds an extra capacity to map making.
4 Versioned mapping

A map is defined by the set of parameter values, dimensions, and the set of values of the dimensions defines the context. When the context changes, that is when the value of one or more dimensions changes, another version of the map is defined. From a formal point of view, the context is a point in a multidimensional space. Changing the value of one or more dimensions leads to a different version of the map, another point in this multidimensional space. For example, changing the value of dimension language from Spanish to French, changes the context to be another point in the space and defines another version of the map, the French version.

Intensional Programming [3] is a general model that can be applied to deal with the variance in any system, object, network, once these are being parametrized (expressed by a list of parameters). Through the use of contexts, this variance is monitored and acted upon where needed.

Maps are characterized by very high variance. They swim in a sea of variables and Intensional Programming is the best way of manipulating many variables at the same time.

5 Implementation

To create a multilingual interactive mapper using the mentioned tools the following steps are needed:

- Develop a user interface to capture users’ inputs and needs, and to display results.
- Add multilingual capabilities to mapping.
- Connect everything to create the mapping server.

5.1 Interface

The user interface is a Web page, which implicitly means that the server can be accessed remotely. It is implemented in ISE (Intensional Sequential Evaluator), a Perl-like language, which handles context-aware entities and context strings.

The page has the capability of adjusting itself to users’ requests. At present it can adapt to the language chosen by the user. This means that the page will reload with the newly chosen language and every new page loaded will bear that language, until changed again.

Other parameters will be added in the future, like level of expertise and level of detail. The level of expertise corresponds to the knowledge the user has about map making and how familiar she is with the system. For beginners, extra explanations will be given or more help buttons will be provided. For expert users certain map parameters will not appear or others will require more detailed input. Level of detail is associated with how exact the user needs to be. Are graphical dials good enough or do numeric values need to be displayed? For Omega users, an option could be given to specify in greater detail how to display the specific language chosen, and other options.

Another function of the interface is to display the newly generated map, along in the Web page that will accept new entries.

5.2 Multilingual capabilities

To introduce multilingual capabilities to the mapping server we need to create a new function in GMT called psomega to replace the function to plot text on maps; and implement a database to handle multilingual place names.

5.2.1 psomega

A new function to plot text strings on maps using Unicode was to be integrated to the mapping tools. The way GMT produces a map is by creating PostScript layers, with a beginning layer, in-the-middle layers, and an ending layer. psomega was designed to create a single layer to interleave with the others.

The input of psomega is the same as for the existing function in GMT, pstext: a list of (longitude, latitude) pairs with associated strings, but for psomega, encoded in Unicode. Using the help of another function, geographical coordinates are converted into PostScript coordinates. Omega is called to plot each pair and string set. Dvips is called to convert to PostScript and a header and footer are added to the layer depending on the placing of the layer in the whole image.

5.2.2 Database

The schema design was to allow expansion, and yet be efficient for data search.
There are three kinds of tables. First, the language table contains all the languages available in the database. One line per language, each line states the name of the table that corresponds to that language. Second, the location table contains the code and the bounding box of the geographical entity. The third kind of tables contains the code and name string of each entity, one table per language.

When a request is made, the language table is searched to verify that the language in the context (the language requested to plot the map) exists in the database. At the same time the location table is searched to find the codes of entities that fit in the region of the map (again region taken from the context). Once the codes are retrieved, the language table chosen is searched to retrieve the Unicode strings and bounding box coordinates of all codes retrieved previously.

5.3 The system

This is how the parts work together. The user enters a request of change; it could be an interface parameter, or a new map with different values. In the previous section I explained what happens with language change, which is the only interface parameter implemented so far. When a new map is requested, the value of the labels-state is checked to give control to the label placement mechanism. This part of the server will check the context for the geographical region and the language of the map values and with them, search the database for the Unicode string to plot and the geographical bounding box as described before. The placement mechanism calculates the point to place the label, in this case the center of the bounding box. With the (longitude, latitude) pair, getGeoFromXY is used to obtain PostScript coordinates. With these pairs and the name string, a file is created in the format needed for psomega to run.

GMT functions are called to make the map. pscoast is required to draw the basemap following the values in the context: geographical region, kind of borders, colors, etc. psomega is called to create the following layer with labels. Not implemented yet, psxy will be used to create an extra layer to plot symbols for cities. The output is a PostScript file, the map, which is converted to png format displayable through the browser. The Web page is displayed with the new map and with all the values as the context has them. The context can now be changed again for a new version to be displayed.
6 Further work

Now that the need for Omega in multilingual mapping and the advantages of handling maps from an intensional point of view are shown, more sophisticated storing mechanisms must be developed. We need real multilingual, multi-naming databases of places around the world. The schemas used at present cannot handle validity of data according to dates, or different names to the same place. Versioned databases must be developed, to handle these or other problems that arise with labeling problems.

At the same time, the Web interface needs to be enhanced to accommodate more of the users' need. Level of expertise, level of detail and Omega options are some of the enhancements scheduled. Map projections also need development, since at the moment only one projection is working. Omega should also be fed more information on what is required (i.e. horizontal or vertical script) and with new language primitives for it to do the best possible job.

References


