

A New Approach to Interactive On-line Mapping

Blanca Mancilla

School of Computer Science and Engineering

University of New South Wales

UNSW SYDNEY NSW 2052, Australia

bmancilla@cse.unsw.edu.au

Abstract: On-line geographical mapping and the widespread availability of mapping software are revolutionizing the world of cartography: anyone can create maps of anywhere in the world. Typically, however, these tools only allow one to produce, with limited choice, static pictures in which the process undertaken is discarded.

I present a new approach to cartography in which maps are defined as the visual representation of a multiparametric (or multidimensional) context. Such a context can be understood to be the environment in which the map is built; it is defined by the values of the parameters used in the process. This approach to mapping — called *intensional*— allows map developers to keep track of all the choices that are being made, whether visual, structural or socio-historical, and to share these choices with others, in real-time. As a result, electronic maps become much more flexible and versatile, and much more supportive of speculative research.

The specific purpose of this thesis is to develop the formal theory for the manipulation of multiple parameters and, from there, to implement an intensional mapping tool with interactive and multilingual capabilities.

Keywords: Interactive mapping, on-line mapping, versioning, intensionality, GMT, multilingual mapping

1 Introduction

Electronic map building requires manipulating different kinds of parameters, independent or interrelated. The efficient manipulation of parameters should allow the creation of more complex maps, and the infrastructure for more powerful mapping tools. Today's tools, however, only handle parameters in an *ad-hoc* manner, allowing limited modification or expansion of existing systems, and little adaptability to changing environments. As a result, users' demands are hard to meet and in the worst of cases totally ignored. These demands include map update, correct data display and process backtrack. Map update is required when a parameter contributing to a map changes and a new instance of the map is to be produced with as little effort as possible. Correct data display is expected when zooming in and out. At each time, the user wants to see the right type of data, of correct magnitude. Technically this means looking for the most suitable data source. Process backtrack is needed when the user wants to revisit previous results

or study how they were obtained. Users are typically faced with a graphical end result detached from the process that led to it. Is a map just an image or is it the result of a complex workflow?

Correctly handling these demands requires a formal framework to handle multiple parameters, called the *Intensional* framework. A map is understood to be immersed in a sea of parameters, a *context*, and each instance, or *version*, depends on the values of these parameters, or *dimensions*. When this context changes—that is, when dimension values change or a dimension is added or deleted—a different version of the map is produced.

2 The Intensional Framework

The ideal mapping environment could in theory include an infinite number of programs and datasets. Changing a parameter could then affect several programs or datasets, possibly in interdependent ways. Rather than having a centralized version controller explicitly specifying the

allowed parameters, and individually notifying each and every program and dataset as changes occur, a more flexible but efficient version update mechanism should be used.

Tools and datasets should *sense* the changes to those parameters that they consider to be essential and should adapt accordingly. The intensional model allows change of structure without having to re-implement the whole system, which would be required by a centralized version controller. In this model, a new object can enter the system and *sense* the values of the existing parameters and interpret them as it wishes. As the context changes, with a new parameter, or a change of value, each object changes without a central system having to notify each object explicitly.

Intensional programming (Plaice & Paquet, 1996) assumes the intensional model. Based on the possible-worlds semantics of intensional logic (Wadge, 2000), it assumes that programs run in an implicit context, a point in a multidimensional space. This space can in theory be arbitrarily complicated, but a simple Cartesian space, where the dimensions and values are simple scalars (strings and integers), is often enough.

Formally, an intensional program is understood to be a function from the multidimensional space to the domain of ordinary programs. Informally, the program is understood to be immersed in the context. When it needs to, the program can test the current values for one or more dimensions and choose alternate paths according to these values.

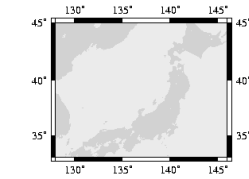
Languages based on intensional programming (Brown, 1998; Swoboda & Wadge, 2000) have been created and used to build a number of interesting applications, including multilingual distance education and multidimensional documents. They are easily applied to other situations, and it is conceptually not difficult —although probably not easy in practice— to add intensionality to all sorts of applications, including file systems, protocols and operating systems (Plaice & Kropf, 1999). In addition, there are means for sharing multidimensional contexts among processes, using what are called *aether servers* (Plaice et al., 2000).

3 The Map's Context

Using GMT (Wessel & Smith, 1998),[†] I have created an interactive mapping tool for the Web,[‡] using CGI scripts in Perl and HTML forms. Figure 1 shows a snapshot of the interface. The map produced at the

top is based on a series of dimension values seen in the rest of the figure. These dimension values specify the context of the map and can formally be written as:

```
Context1 =
proj:Miller + lonMin:128 + lonMax:146 +
latMin:33 + latMax:45 + landColor:210 +
waterColor:235 + resol:low + anotSpX:5
```



GMT 2000 Nov 29 10:31:12

Let's make the map Clear

Map Projection <input type="checkbox"/> Mercator <input type="checkbox"/> Cassini <input checked="" type="checkbox"/> Miller <input type="checkbox"/> Equidistant <input type="checkbox"/> Stereographic <input type="checkbox"/> Orthographic <input type="checkbox"/> Hammer	Choose one projection and keep in mind the following: Mercator, Cassini and Miller are cylindrical; Stereographic, Equidistant and Orthographic are azimuthal. <i>Mercator</i> is good for about anything except the poles. <i>Cassini</i> is best suited for mapping regions of north-southeast. <i>Miller</i> is a compromise between the Cassini and Mercator. In <i>Orthographic</i> , only one hemisphere can be viewed at any time. <i>Stereographic</i> is mainly used for mapping the polar regions. <i>Equidistant</i> is useful for a global view of locations.
Region 128 Min longitude 146 Max longitude 33 Min latitude 45 Max latitude	Must choose a region <i>Longitude</i> must have a range of 360 degrees. The lower limit is -180 degrees and the upper limit is 360. To specify a central meridian of say 0 degrees, the limits should be -180 and 180. <i>Latitude</i> limits are -90 an 90 degrees where 0 represents the Equator.

The following parameters are optional, they make your map prettier and understandable.

Label Information Tick information and plot labelling <input type="text"/> Plot Title	
Info X-axis <input type="text" value="5"/> Annotation spacing <input type="text"/> Frame spacing <input type="text"/> Grid spacing	Info Y-axis <input type="text"/> Annotation spacing <input type="text"/> Frame spacing <input type="text"/> Grid spacing
Land Color Components between 0 and 255 <input type="text" value="210"/> Red component/Grey <input type="text"/> Green component <input type="text"/> Blue component	Water Color Components between 0 and 255 <input type="text" value="235"/> Red component/Grey <input type="text"/> Green component <input type="text"/> Blue component

Range of all values is between 0 and 255. If only the first value is included, it is taken as a grey level.

Choose the type of boundary

Author

Let's make the map Clear

- Resolution
- Crude
 - Low
 - Intermediate
 - High
 - Full

Figure 1: Simple mapping interface

[†]Generic Mapping Tools, <http://www.soest.hawaii.edu/gmt>

[‡]<http://www.cse.unsw.EDU.AU/~bmancilla/mapping/mapping.html>

As far as this interface is concerned `Context1` fully specifies the map seen in the figure. Any other parameters are fixed internally. This map contains one layer produced by `pscoast`, a function of GMT, and `Context1` refers to dimensions used by `pscoast`.

Figure 2 contains a map of Europe with some capital cities and outdated border information (the former Yugoslavia appears as one country). This map was created using three different functions in GMT version 3.3.4, namely `pscoast` for borders, `pstext` for labels and `psxy` for city dots.

Each function creates a layer and all layers are immersed in the same context. Dimensions that do not affect a function are ignored by that function.

The context to describe this map is as follows:

```
Context2 =
proj:Cassini + lonMin:-12 + lonMax:41 +
latMin:35 + latMax:65 + anotSpX:10 +
anotSpY:10 + subContext2
where subContext2 =
landColor:230 + waterColor:250 +
natBorders:on + outputFile:map.ps +
labelsFile:labelsEN + dotsFile:dots +
orientation:portrait + version:3.3.4
```

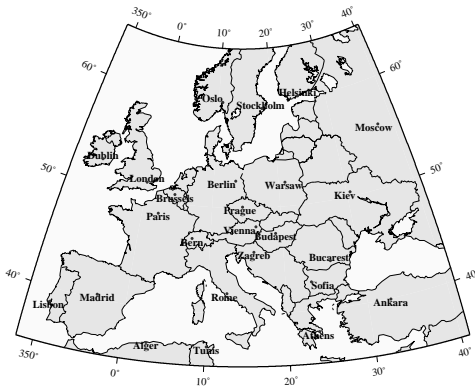


Figure 2: Europe with national borders and capital cities

Figure 3 is a zoom in of Figure 2 and is described by the following context:

```
Context3 =
proj:Mercator + lonMin:0 + lonMax:35 +
latMin:40 + latMax:55 + anotSpX:5 +
anotSpY:5 + subContext2
```

Now let's assume that an update for borders is introduced to the system (GMT version 3.3.6 is installed). The next time that either `Context2` or `Context3` is consulted following a user request for the latest version, the dimension `version` is set to 3.3.6 and the updated map will be rendered. If

the user wants to see the older borders, `version` can be set back to 3.3.4 for rendering. From this, map update occurs through a change of context, not through a change of data. No information is deleted or modified, just new data added. Older data becomes *history* that can be made accessible, through requests such as *Give me a map of Europe with national borders as of 1990*.

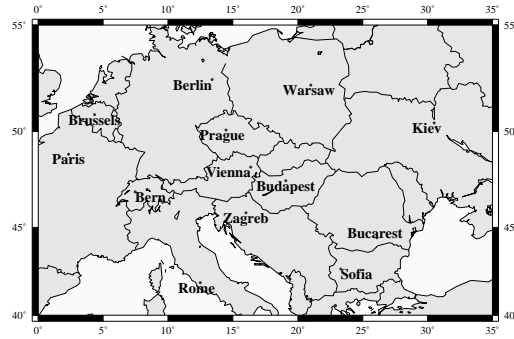


Figure 3: Central Europe, zoom in of Figure 2

3.1 History Logs

With the notion that a context completely describes a map, and that a user can iterate through a series of maps before arriving to a satisfactory state, we can see that the series of visited contexts is a *history log*. A result map is then the final point in a sequence of contexts; it is the workflow that led to it. The log allows the user to answer the question, "How did you get to this point?" Furthermore, the user should be able to *annotate* sections of the workflow. For example, from `contextA` to `contextB` the user applied a filter, $f(x)$. The user could annotate the filter by adding a comment on why this decision was made and explaining technicalities.

3.2 Databases

A user should be able to zoom in on a map and, as the level of detail changes, the most appropriate dataset should be used. When viewing a map of the world, for example, the objects represented by displayed map objects should correspond to the available map space in order of object importance[§]. After zooming in to city level, the data of higher detail should be available: streets, train routes, parks, buildings, etc. If the process continues all the way down to house level, the plan of the house should appear. The

[§]If the objects are cities, capital cities are the first priority (if space still limited, they might have an internal ranking). If space is available, then other important cities are shown, and as zooming in continues, smaller cities and towns show up, an so on.

infrastructure should allow for dynamic searches of data and adjust to use the *best-fitted* source.

3.3 Collaborative Work

With the concept of context logs, the notion of *sharing* becomes a real possibility. Sharing here refers to collaborative work, people getting together with a common objective or to share points of view for discussion. This sounds like the sort of thing that should happen in any research environment, although typically it only happens at the personal level, with the computer as just an annex. Shouldn't people in different corners of the world be able to collaborate in cyberspace on a project, or discuss ideas and points of view? In other words, why not use the computer to facilitate interaction between people, regardless of their geographical locations?

Let's refer specifically to mapping. The mapping infrastructure previously described allows different users to contribute to the same context to build a map together. Or one user can lead a tour through a series of contexts while the other users follow, a situation similar to students following lecturer notes.

Using parametrization—the use of dimensions—users can adopt different views while still sharing a common subcontext, thereby facilitating speculative research. If only points of disagreement stand out, discussion is clearer. Using the recorded session log, users can collectively go through the process and explain each step taken.

4 A New Definition of a Map

An electronic map is a flexible document that allows the user to manipulate its structure as well as its content. An electronic map should be a family of versions of maps and the entire collection should be perceived as one entity. The behaviour of a map is dependent on the values that these dimensions hold and they fully define its structure, its content, its output representation, its computational environment. As the map becomes a more complex entity supporting a more complex set of features, so do its dimensions, their relationships and their manipulation. As multidimensional entities, maps need tools based on the formal theory for multidimensionality, namely intensional tools.

5 Short Term Goals

The goal of this PhD thesis is to define and implement the infrastructure to intensionally manipulate electronic maps. At the practical level this means to create an intensional interface between the user and

the mapping tool to allow the dynamic manipulation (add, delete, modify) of the many dimensions required for the production of maps. Specifically for this project, the dimension `language` will be added and `levelOfDetail` will be modified, affecting more than just GMT internal dimensions. This will require the existence of a multilingual database of names and the development of an intensional query language (iSQL), to extract the appropriate data relevant to the right level of detail, language and map scale for the mapping engine.

Some of the difficulties will arise from the implementation of the iSQL for different types of databases and diverse requirements. Introducing a new layer of a GMT output could also present a problem, since multilingual support in GMT is limited. It would be necessary then to create an extension of the GMT `pstext` function using a mature typesetting system called Omega,[†] designed to handle all of the world's scripts.

6 Conclusion

Viewing maps through an intensional lens is very promising. Already this limited prototype for interactive mapping allow users to do a number of tasks that might only be doable with the most expensive commercial tools. As the ideas in this paper are developed, research in any discipline that requires any kind of maps will benefit substantially.

References

- Brown, G. D. (1998), *Intensional HTML 2: A Practical Approach*, Master's thesis, Department of Computer Science, University of Victoria, Canada.
- Plaice, J. & Kropf, P. (1999), *Intensional Programming II*, World-Scientific, chapter Intensional communities.
- Plaice, J. & Paquet, J. (1996), *Intensional Programming I*, World Scientific, chapter Introduction to Intensional Programming.
- Plaice, J., Swoboda, P. & Alammari, A. (2000), *Distributed Communities on the Web*, Springer, chapter Building intensional communities using shared contexts.
- Swoboda, P. & Wadge, W. (2000), *Intensional Programming II*, World-Scientific, chapter Vmake, ISE and IRCS: General tools for the intensionalization of software systems.
- Wadge, W. W. (2000), *Intensional Programming II*, World Scientific, chapter Intensional Logic in Context.
- Wessel, P. & Smith, W. (1998), "New, improved version of Generic Mapping Tools released", *EOS transactions* **79**, 579.

[†]<http://omega-system.sourceforge.net/>