ISOLA project environmental information system

Andrea Aime e Silvia Ascari(*), Flavio Bonfatti e Paola Monari(**)

(*) Comune di Modena, (**) Università di Modena e Reggio Emilia

Introduction

The ISOLA project, funded in the framework of the European LIFE/Environment programme and co-ordinated by the municipality of Modena, is aimed at providing methods and tools to support environmental control and planning for a medium-sized town. The expected project results are:

- a sound methodology, expressed in form of guidelines addressed to the end users that plan to adopt a disciplined way of facing the environmental problem.
- a software package to support the methodology and make its application directly feasible and controlled by end users not expert in GIS technology.

From the methodological point of view, ISOLA is based on three main activities:

- Eco-balance. Assessment of risks in the urban environment and its components, and measurement of the pressure exerted by human activities and settlements.
- Eco-plan. Integration of environmental factors in urban planning according to proper methods, criteria and procedures.
- Eco-management. Support to decision-making by means of simulation tools that allow evaluating the impact of alternative scenarios.

For a more in-deep treatment of the methodological issues refer (1) and the ISOLA web site (2).

Every methodological activity requires some kind of spatial analysis, these activities need to be carried out as a result of the collaboration several experts in different environmental issues, and are carried out on a periodical basis: this means that the software system, named SIA (Sistema Informativo Ambientale, that is, Environmental Information System) has to put a strong focus on spatial data analysis and representation, information sharing and analysis automation. Since ISOLA project is funded by a Europeand programme results have to be useful for every other medium sized municipality, thus exportability of software, in terms of cost, technology and automated activities is a major concern.

In this paper we outline the main problem that SIA has to face and the solutions found in order to effectively face competing and apparently incompatible requirements. Section 2 provides a more in depth analysis on the issues that a spatial analysis system has to face in order to be versatile, user friendly and effective. Section 3 describes the process model, a graphical tools providing a good compromise between automation and user friendliness. Section 4 describes the client server architecture that enables user to share both data and processes in a cost effective way by leveraging as much open source software as possible, and in particular how GRASS (3) is being used to provide the most important foundation to environmental data analysis, while Section 5 draws conclusive remarks and gives some hints about future improvements.

Spatial analysis development and related issues

By spatial analysis we mean every kind of transformation that takes some spatial data as input in order to provide new synthetic information that can be used to support decision-making.

The development of a spatial analysis can be thought as a stepwise process that is carried out in an incremental and iterative fashion. As with every modelling activity, we usually start with a rather weak comprehension about how to address the problem and end up with a process definition that solves the problem and faces its most common pitfalls (see Figure 1).

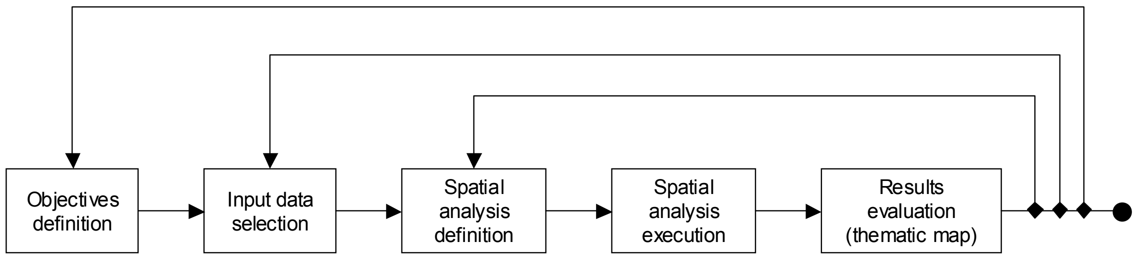

Figure 1 – spatial analysis development process

As we can guess by looking at Figure 1, the ability to store the sequence of commands that makes up the anaysis process is key to quickly approach the definitive result.
Most spatial analysis processes needs to be carried out on a periodical base or to be applied to some other region, so it's also useful to store the steps and have them issued again in a automated fashion.

Although there are common problem, municipalities have peculiar environmental issues, related to the local environment features, different levels of data availability and maybe different environmental objectives, so we expect some spatial analysis to be developed ad hoc to face some peculiar problem, and some other to be adapted to different objectives or different data available. Furthermore, analysis may change as result of change in legislation or objectives, and some analysis may be created and executed only to face a contingent problem. We can draw some conclusion about the set of analysis that SIA has to support:

- the set may change in time and from municipality to municipality;
- every analysis has to be executed in an automated way;
- the set is a moving target, so it must be easy to create a new analysis or to adapt an existing one.

The common answer to that problem is the development of a set of procedures in a scripting language provided with the GIS software, in particular, in GRASS script can be recorded and developed using the standard Bourne shell. This is not the kind of solution we were looking for, since it is associated with three main problems:

- domain experts are capable of explaining the steps that needs to be carried out in order to make some kind of spatial analysis, but they usually aren't able to use directly the GIS engine (this holds true in particular for GRASS, with a command line interface under a UNIX environment);
- a GIS expert with software development skills is needed to create and modify such a script, furthermore the different culture between domain experts and GIS experts leads to a increased number of iteration during analysis development;
- the script is formalized and provides the best documentation about how the analysis is carried out, but it is difficult to read from the domain expert point of view.

We were looking for a solution that gives the end user the power of a scripting language without the need to understand the technical details related to it: we called that solution the process model.

The process model and the process editor

A process is intended as an ordered sequence of operations aimed at manipulating spatial data with the purpose of extracting additional meaning as a result.

Data, operations, and links are the basic process concepts:

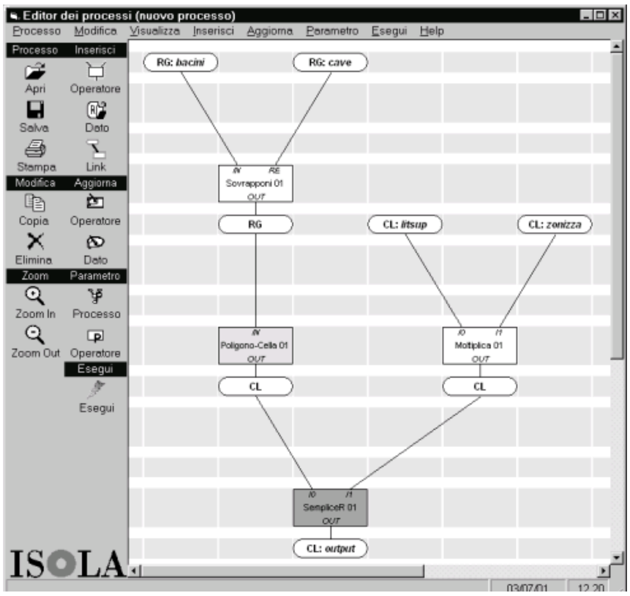

Figure 2 – process editor with a sample process

- data are spatial information manipulated by the process, each one associated with a type, composed of a geometric characterisation (point, line, polygon and raster) and a set of attributes (only one for raster data, zero or more for vector data). Each data can be persistent or intermediate, the latter meaning that it will be thrown away as the process terminates its execution;
- links represent input/output relations between data and operators, in other words they connect operators to their input and output data;
- operators represent the spatial operation performed on available data in order to derive new information. Each operator has a set of input roles (each one with a specific meaning) that will be linked to data, a set of parameters to decide how the operator computes the output data, and of course a set of output data. Our current prototype support both raster and vector operations, as well as conversions between raster and vector format (including simple interpolation methods).

Spatial analyses processes are developed by means of a process editor, an application allowing for interactive building, validation ad execution of processes.

Validation means that the process editor is not a simple flowcharting tool, since it assists the user in the creation of correct and executable processes, by means of semantic controls and user feedback by means of colour codes. The editor is able to detected operators that lack inputs, that lack parameters (such as the buffering distance for a buffering operation) and operators that are connected to unsuitable input. Controls are executed in real time, as the user creates and modifies the process, allowing for an immediate evaluation of the effect of a change on the whole process.

Editor provides also means to launch process execution, get feedback as each operator is executed and prematurely stop the execution of a time-consuming process.

Process modelling by graphical means is not new, and we can find some other examples of this kind of application, in particular ESRI “model builder”, included in the spatial analyst extension module (5), and Idrisi32 “macro modeler” (6) (both appeared on the market after ISOLA project was conceived).

Information sharing

The information system is based on a client/server approach, thus enabling information sharing among different users by means of a shared repository, called the Virtual File System. The name implies that the users perceive the VFS as a standard file system, and that the physical structure is somewhat different from the one that the user experience.

In fact, being based on GRASS, the VFS is a set of locations and mapsets, but the users perceives them as a single file system in which every user gets a home folder, plus a common folder that every user can access to store shared information, and every home folder can be organized in sub-folder at the user's convenience.

Every user has to authenticate himself at start-up and, depending on the access level, he may have write grants limited to its own folder, or may be enabled to write to the common folder. This prevents untrained people to damage important analysis processes while letting them try process development on their own home folder.

Every entity managed by the system, being it a process, data or a thematic map definition, is perceived as treated as a single file, independent of its physical nature. In the ISOLA prototype, most information is stored in the file system, but with a different structure than the one perceived by the user, while other information is stored in a centralised database (mainly metadata and vector data attribute tables).

Users interact with the VFS by means of the SIA explorer, a MS Windows explorer-like interface that lets the user do common file operations (move, copy, rename, delete and folder creation) as well as import and export activities.

System architecture

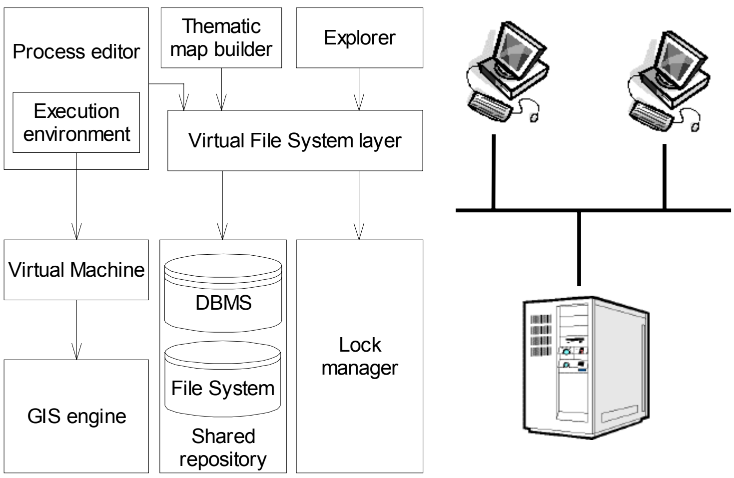

Figure 3 – system architecture, main modules

SIA general architecture is based on a traditional client-server architecture, where the client has in charge the interaction with the user and the server manages persistent data and executes computations. This separation ensures the independence of the client functions from the underlying GIS technology, and allows implementing the ISOLA environment on top of different GIS platforms.

The client side of the architecture includes the process editor, the thematic map builder and the explorer. The server side includes the DMBS, one or more GIS engine in charge of spatial analysis, a virtual machine that allows for GIS engine independence and a lock manager that is used to coordinate concurrent access to shared resources on the file system.

For what concern the use of DBMS, independence from the vendor is obtained by using only SQL standard features and by using access library that are built to assure client independence from the particular DBMS (ODBC). In our current implementation we are using PostgreSQL (6), an open source DBMS that provides full SQL compliance, transaction support and an ODBC driver that let Windows clients access data in a standardized way.

Process execution is made independent of the GIS engine by using a virtual machine, that is, a software component that exposes a standard interface to the process execution environment and translating requests in direct API call to the GIS engine. If GIS engine has to be changed, only the virtual machine needs to be rewritten, preserving all the code that manages process construction and user interaction. Process execution is performed in two steps:

- First, the execution environment built into the process editor translates (compiles) high-level operators into calls to a virtual machine, whose interface is still independent of GIS engine.
- Then, the virtual machine translates this intermediate language into specific GIS engine calls.

Since the system is build to allow concurrent use, this feature is exploited to allow for concurrent execution of operators contained in a single process, in a way that allows to take advantage of the full processing power of the server side.

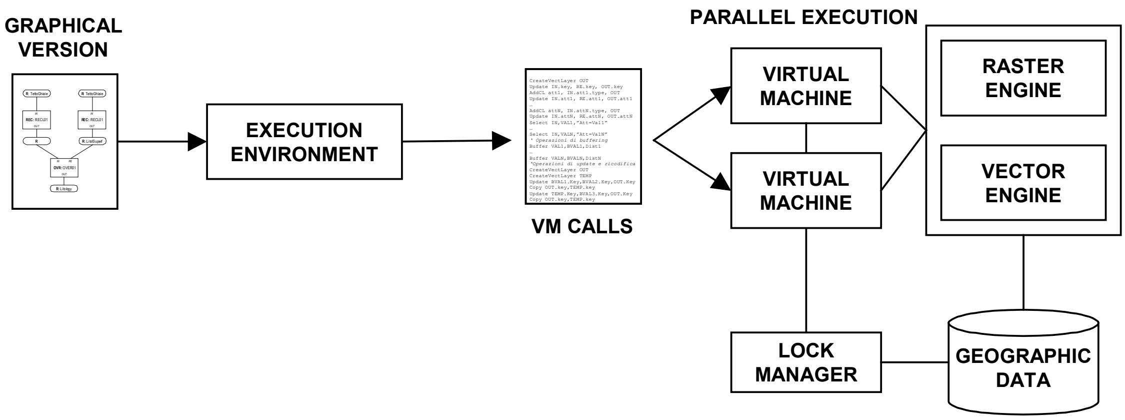

Figure 4 – process translation and execution

Taking into consideration our current prototype, GIS execution is performed both on the client side and on the sever side: this mismatch is due to the lack of vector analysis capabilities in GRASS, that force us to perform such analysis on the client side using Autocad Map, that is also used as a thematic map builder (an extension allows us to read and render GRASS raster data). Improvements in GRASS vector capabilities will allow us to restore full client/server architecture.

Conclusions and further improvements

Summarizing, the main SIA benefits are:

- SIA faces problem related to spatial analysis evolution, migration and automation by a good compromise between a rigid full encapsulation and a versatile but too technical script-based approach.
- System architecture provides for efficient hardware allocation, in that most of the heavy computation task are located on a centralized machine, and information sharing, allowing several different domain experts to share both data, analysis processes and thematic maps.
- Process execution layering allows for GIS engine change.
- The standard prototype is based on open source software, allowing for a cheap spatial analysis system.

SIA modular architecture allows for several future improvements:

- support for non spatial data, in particular for tabular data and time series, both on the side of data analysis and on the side of data visualisation (by means of textual reports and charts);

- support for remotely sensed data, again both on the processing a visualization point of view;
- thematic maps are stored as a structure definition, making them usable to support different kinds of viewers and to allow for integrated Web mapping. The user could be enabled to publish thematic maps on the web by simple dragging thematic map definition into a special folder. A reference Web mapping tool for such an activity could be MapServer;
- more complex system could be integrated into SIA, such as simulation models.

References

(1) F. Bonfatti, P. D. Monari, C. A. Muratori. *The ISOLA project: a novel approach to urban environment data use*, UDMS '99 International Symposium, Venice, 1999.
(2) The ISOLA web site, http://www.comune.modena.it/isola
(3) GRASS european web site, http://www.geog.uni-hannover.de/grass
(4) ESRI ArcGis Spatial Analyst, http://www.esri.com/software/arcgis/arcgisextensions/spatialanalyst/index.html
(5) Clark Labs Idrisi32 release 2, http://www.idrisi.com/03prod/03prod.htm
(6) PostgreSQL web site, http://www.postgresql.org