

A Java Based Framework Optimized for Scientific Modeling and Analysis

P. Cau, G. C. Meloni, S. Manca, C. Soru, D. Muroi

Abstract—Environmental sciences are moving from a simple, local-scale approach toward complex multilayered, spatially explicit regional ones. The new paradigm is based on integrated and collaborative web tools where the complexity of the technology is transparent to the end user, and interdisciplinary working groups and skills can be enhanced. In this context scientific portals are becoming strategic gateways where end users and stakeholders can securely use innovative applications and researchers and scientists can transparently access to data, computational infrastructures and services.

Development frameworks intend to simplify development and integration of such web-based, service oriented environments. BASHYT is a Java platform, based on the model-view-controller (MVC) architectural pattern, to design GIS oriented, Web Information Systems (WIS). The software exposes modules for temporal and spatial (graph, GIS, etc.) analysis to support the dynamic, real time, report production mechanism. At current state, the open source hydrological SWAT and GETM oceanographic models have been interfaced to the BASHYT environment. Our aim is to build an experimental programming platform to run real-time applications based on environmental numerical solvers, run pre- and post-processing codes, query and map results through the web browser. We expect to improve WIS development and maintenance and to improve model usability to address more realistically environmental management. To illustrate the potentiality of the system, we present its use in the EnviroGRIDS and MOMAR projects.

Index Terms—BASHYT, Web Information System, environmental models, web development framework, Spatialite.

I. INTRODUCTION

THE latest advances in computer science, high performance computing and web based technologies have highly extended the possibilities in the environmental

P. Cau is researcher for Advanced Studies, Research and Development in Sardinia (CRS4), <http://www.crs4.it>, Pula, , CO 09010 Italy (phone: +39 070 9250281 - +39 3397745578 - fax: +39 0709250216; e-mail: pierluigi.cau@gmail.com).

S. Manca. is researcher for Advanced Studies, Research and Development in Sardinia (CRS4), <http://www.crs4.it>, Pula, , CO 09010 Italy (phone: +39 070 9250236 - fax: +39 0709250216; e-mail: simone.manca@gmail.com)

C. Soru is researcher for CRS4, <http://www.crs4.it>, Pula, Italy (phone: +39 070 9250346 -fax: +39 0709250216; e-mail: costantino.soru@gmail.com).

G.C. Meloni, is head of the ERA Progetti development team – Cagliari, Itali, V. Vincenzo Monti 33. (<http://www.eraprogetti.com> - eraprogettimit@gmail.com).

D. Muroi is researcher for CRS4, <http://www.crs4.it>, Pula, CO 09010 Italy (phone: +39 070 9250361 - fax: +39 0709250216; e-mail: davide85@gmail.com).

sciences, and have changed the ways in which information systems operate, providing important services, applications and advanced visualization tools. Web applications combine and use complex data infrastructure, execute models, process and interpret input and outputs, and retrieve analysis, exposing important services to the WEB [1,2]. Such frameworks offer a uniform way of identifying and accessing to resources, and thus increasing the interoperability between applications. Web applications are mostly data-driven, and it is easy predictable the increasing interoperability and data reuse through “mashups” that merge information, model outputs, or simply territorial data from various Web data providers.

Earth Science and environmental agencies are looking ever more for scientific portals as centric gateways to applications and services, based on workflow and dataflow mechanisms and as a strong support in accessing to quality information. Although such portals are key components of scientific research and for reporting systems mechanisms, their development requirements are hardly met by current development platforms. In addition, it is still a very common practice to use desktop tools to exploit data and processes, even though scientific workflow tools could make use of web services to access resources and services. Traditionally, desktop solutions are designed to do the actual computational jobs while the website is used mainly to show results.

Recently, web development frameworks such as Ruby on Rails (<http://rubyonrails.org>) or Django, which adds also a loosely-coupled, high-level Python interface for GIS geometry operations and data formats, (<http://www.djangoproject.com>) have imposed more structured and conceptualized ways to design web applications. Such frameworks are based on the MVC design paradigm [3,4] and efficiently enable to shape web applications, solving the problem of the division of the application logic for the user from input and presentation. These frameworks offer only limited support for interoperability. Usually the design process is done in a initial phase and once the application is exposed on the web can hardly be changed. In such a context, the development of web applications for the environmental sciences is very much in the hands of the software developer, rather than the scientist due to the complexity of the task.

On the data level, web applications use and produce large amounts of data and neither the choice of a traditional RDBM's or a data store infrastructure solves all pros and cons of operational requirements. The design is usually driven by application workload, data accessibility, scalability and preexistent data configuration. RDBMS's are,

in general, not scalable and force data to be twisted to fit into the relational world. On the other hand, data stores are characterized by their lack of referential integrity, transaction support and data consistency. Experience shows that their use is only realistic when developing applications from scratch.

The BASHYT is a java based, development framework to design complex, data driven, GIS oriented web applications optimized for scientific modeling and temporal - spatial enabled analysis. The platform is based on a distributed DBMS paradigm, a compromise between the traditional RDBMS's and a file based distributed data store architecture. The aim of the BASHYT technological framework is to support and encourage scientists to develop their own web application from scratch, maintain and further develop new services. The software exposes on the web (Wiki like) a fast and flexible processing system for service management and development making the programming features available with almost-zero learning curve.

II. THE JAVA FRAMEWORK

BASHYT can be thought as an easy to use and extensible development framework for constructing spatially enabled web applications. In the back end, BASHYT exploits the Argilla engine, a self consistent development environment for generic WIS development supported by the ERA Progetti srl (<http://www.eraprogetti.com>). It exploits also the MVC architectural pattern enabling for each component, independent development, testing and maintenance. The system makes intensive use of complex server side technologies and easy to use client side interfaces using an approach founded on centralizing all model related data into a complex Relational DataBase infrastructure. Differently from other approaches, data access scalability is obtained accessing a distributed spatialite (<http://www.gaia-gis.it/>) db file environment.

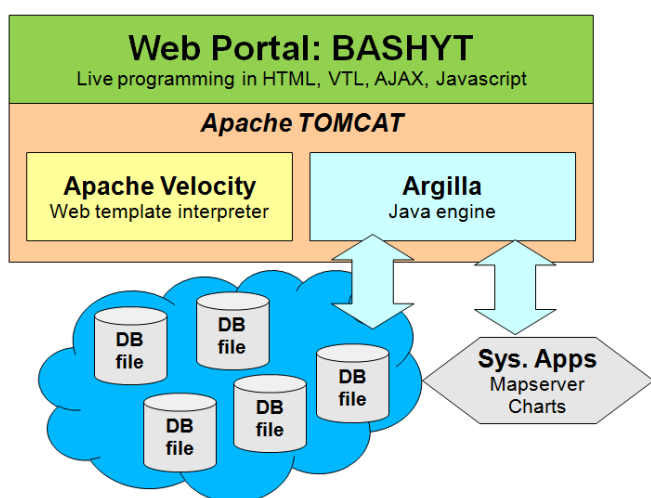


Fig. 1. The MVC paradigm permits easily to create objects such as: Tables, Charts, Forms, Layers, etc. These are created filling up XML modules using different schemas and then exposed in the web interfaces.

The system has connectors to environmental models such as GETM [7,8] and SWAT [9]. Each code simulates 3 dimensional fields of environmental variables over time.

Simulations are submitted in a dedicated computing environment where automatic Extract, Transform, an Load (ETL) procedure process input and output to produce Spatialite db files. SWAT is a watershed-scale hydrological model, developed by the U.S. Department of Agriculture USDA-ARS and Texas A & M University, which allows to simulate the integrated water cycle and to assess the impact in the medium and long term of point and diffuse pollution. The application of the model requires specific information on weather, soil characteristics, topography, vegetation and land use. General Estuarine Transport Model (GETM) was created to be applied in the shelf seas with relatively large tides, where the vertical mixing is intensive. It proved to be rather useful in estuaries when studying the mixing between fresh river and sea waters. GETM simulates the most important hydrodynamic and thermodynamic processes in natural waters, like currents, temperature, salinity, sea level, vertically integrated water transport, turbulent mixing characteristics, water density.

The GIS rendering is optimized integrating Open Source technologies, such as the MapServer for the server side GIS rendering. This is accomplished, using the scripting languages capabilities to access the MapServer CGI and OGC (WMS, WFS) interfaces. MapServer works as a map engine providing a spatial context where it is required. On the client side the AJAX (web 2.0) msCross cross-browser interface [10], is customized and developed to allow users dynamically display and browse the geographical information layers.

The framework permits to write Velocity Templates, which are stand-alone scripts (written in VTL - Velocity scripting language) that combines data such as SWAT simulations, maps, users' roles, to produce a web page (in a HTML format). Applications for the reporting production, in this way, uses the full features of the web browser, so it is possible to integrate JavaScript / AJAX objects in the same developing environment.

A. The Argilla architecture

Software reusability is not fully accomplished if it is not programmed using a rigorous architectural design. We have chosen to use the MVC conceptual model, and the Java development language, where the MVC design principles are fulfilled by the binding to the specific Java Interfaces and Components available. The logic of the Model from the View or the Controller are completely separated. In terms of the web-based approach desired, the View component is fulfilled by means of Java Server Pages, which deliver the requested html to any browser.

The engine allows to integrate several client and server technologies in a single development environment, fully programmable and accessible by the web browser. Differently from other solutions, in our case developers can write server side codes directly from the web, and use the framework tool for debugging and validation. The Velocity Template allows a strong integration with low-level API written in Java, working as PHP does.

All web applications and pages exposed are described in a structured and hierarchical way within the virtual filesystem (scripts, text contents, applications, configuration files, parameters, etc.). The filesystem is

designed to be hierarchical, much like the common physical filesystem used in traditional operating systems. It is called virtual because files, folders and hierarchies are saved in tables of a SQLite relational database file. In such hierarchy, each folder is a node of the portal: each node is accessible from the browser via a specific URL, and contains (virtual) data files such as the Velocity scripts, HTML, JavaScript, which contribute to the composition of the page requested by the user.

For example, when the user calls the URL /apps/example, the framework constructs a response rendering the files in the /apps/example path. Within this virtual directory, a structured system is based on:

- a "body" file: it contains the main HTML;
- a "javascript" file: it contains the data that populates the section on the javascript in the HTML page),
- a "lmenu" file: it contains the menu on the left of the page in XML).

These several files are characterized by a set of metadata, including type, which enables the Model to render the corresponding class. All files may contain constructs and variables of the Velocity Context, which allow to make the system dynamic and homogeneous; Velocity is interpreted by the same engine, and templates share the same environment variables and data.

Writing objects is simplified: the framework allows the enabled user to do so in a transparent manner using the WEB interface commands. Through the browser, each page to be developed is exposed on the web editor where a button bar allows to edit the body, javascript, lmenu files, and to create new nodes of the tree. The framework's modules permit to design new objects (see chapter on modules) and store them in the virtual file system as structured files, in similar way as described above. The architecture described provides an abstraction level such as to build a complete web portal using only on-line visual tools offered by the framework.

The interoperability level to access to data, relational or not, is the core of all information systems, especially of environmental and geographical systems. In our system traditional structured data can be accessed via the JDBC Java interface, which allows access to most existing RDBMS engines. The use of these sources is guaranteed by SQL calls: special classes let you manage the data flows in and out. The SQL statements can be used as source in all objects defined in modules section, for example in the Charts, which can draw time series contained in any RDBMS. However, the new architectural prototype of web information systems are moving to more simple and scalable distributed no-relational repositories (key/value). These paradigms are used for huge infrastructures with thousands of simultaneous requests, and ensure maximum efficiency in terms of scalability and performance.

B. The storage infrastructure

We have designed a new prototype of a scalable distributed Geodatabase based on SpatialLite for large distributed data-intensive, high scalable applications optimized for the SWAT and GETM model. In the present paper only the SWAT dataflow procedures are presented. Data are produced by the environmental model which run on a dedicated computing environment. The storage system is accessed by BASHYT which acts as a work flow manager

posting requests and getting results. In this configuration the computing and storing tasks are resolved outside the framework.

While sharing many of the same goals as other distributed file systems, our design has been driven by observations of our application workloads and technological environment, both present and expected. This has led us to reconsider the traditional choice of one Omni comprehensive PostGIS database (which still keeps its validity when dealing with a limited dataset) and explore radically different design points. Given the amount of spatial data required for our SWAT watershed scale model for a large scale application, we decided to experiment a solution based on the SQLite technology with spatial extension (SpatialLite) providing a large set of spatial functions and data structures. SQLite offers the capability to load personal or third party extensions (shared libraries), written in C or other languages. This mechanism can be used to straighten the SQL functionalities of the engine or override its functions.

SQLite is an embedded database engine distributed as a common library; it is widely used on many popular applications like Mozilla Firefox, Apple Mac OS X, Google Apps and many more.

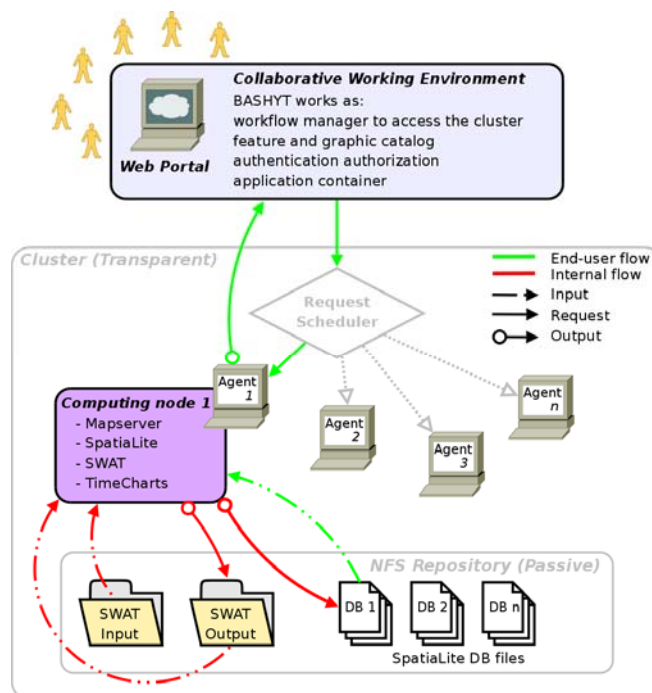


Fig. 2. The computing stage and storage is commanded by Agents to access the distributed filesystem. Data are stored in db files. Each db file contain a model simulation.

Traditional RDBMS's could not be flexible enough to meet the requirement of scalability of a regional/continental context where virtually hundreds of basins need to be simulated. The light weight of the library (~300KB) and the serverless nature of this engine, assures high scalable scenarios, since all operations work as common read/write filesystem calls. This architecture does not need added configuration or administration charges. Linking our application to libsqlite results in gaining the power of a complete transactional RDBMS, without the need of external server process to query and with a useful portability freedom. The main issue to consider when using SQLite is

its strict dependence on filesystem. Often distributed network filesystems suffer from file locking bugs. In general this can cause SQLite data corruption or inconsistency in high traffic volume contexts. In SQLite, one reading operation locks all write requests on files and vice versa; in high concurrency conditions, when read/write actions alternate themselves with high-frequency, this could represent a performance bottleneck. Although our system aims at working in high volume data and traffic situations, the above issues are minor, because end-user operations are read only operations; as a matter of fact all write operations are done batch procedures only to import SWAT outputs. During this task, the simulation is not available to users for reading.

We tested SQLite carefully mostly with regards to its SpatiaLite extension. This technology on one hand can still be considered young and does not have the reliability level or spatial functions of other engines like PostGIS or ORACLE Spatial. On the other hand for a limited controlled use, SpatiaLite meets our needs, although some changes on JDBC SQLite driver for Java were needed to let it work on our distributed system.

C. The Wiki paradigm for developers

Our development platform extends the functionalities of the web Template System. In particular it exposes a fast and flexible processing system on the WEB (Wiki like) for web content management and application development. Earth scientists, through a dedicated web editor, write their own GUI's and applications. The development process, its layout, etc. can be controlled on the fly by switching from edit to view mode. No compilation is required. This increases developer productivity by reducing scaffolding code when developing web GUI, GIS enabled system or any web based application. Hydrologists, scientists, web designers, and developers are asked to concentrate on generating web applications without getting bogged down in programming matters, making the whole process of developing, updating and maintaining web applications significantly easier.

Portal developers are enabled to write services merging server side and client side codes within a uniform development interface. A dedicated section of the development framework exposes modules for the report production mechanism.

D. The modules

The various features for developers are grouped in the module section, where a variety of services enables to shape XML objects for graph, map, table, PDF report, forms production.

Modules permit the massive use of preset schemas stored in the database (virtual file system) in a structured form (XML). Each object refer to its schema and describes parameters (e.g. to control layout) and data sources. The development framework expose a user interface to produce in a easily fashion these objects.



Fig. 3. Argilla exposes the modules section. Through this section web developers can use preset schemas to fill the XML that will be rendered by the Model on the web application

Users are guided by means of preset schemas for the XML data production. Schemas are also used in the validation process, which guarantees the formal consistency of the object parsed. Each user defined object is parsed and dynamically computed anytime a client call is received. This mechanism permits to expose on the client side dynamic rendering services for each object.

III. APPLICATION OF THE FRAMEWORK

The previously cited enviroGRIDS [5] and MOMAR [6] projects strive for the improvement of transnational cooperation while applying innovative, inexpensive monitoring techniques in the Mediterranean and Black Sea basins.

A double objective to be achieved is to set up a complex modeling environment for inland water – marine water management and the integration of such modeling system within a web based technological framework optimised for data management and dynamic report production. The use of state of the art models aim at evaluating the impact of land use, climate and their changes, point pollution sources (civil and industrial compartments) on the river and sea water quality. Both the Black Sea and the Mediterranean Basin are internationally recognized for its ecologically delicate ecosystems where inadequate resource management may lead to severe environmental, social and economical problems. We are addressing these issues by bringing several new emerging web information technologies to build a data-driven vision of our planet that is feeding into models and scenarios to explore the past, the present and the future of these regions.

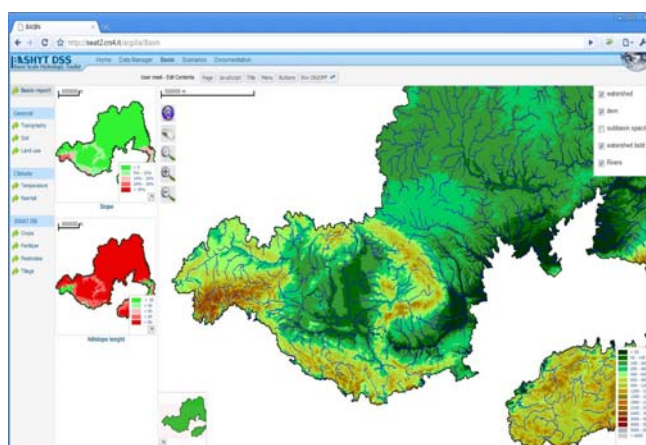


Fig. 4. A view of the Black Sea domain (eastern Europe). A more then 2 million square Kilometer basin is analyzed by web system.

Within the projects, the BASHYT environment is being used to let enlarged working groups collaboratively develop new state of the art web applications.

As an example, we show here a complex environmental application exposed on the portal to analyze drought conditions. Drought is a temporary condition of relative scarcity of water resource compared to values that can be considered normal for a period of time and on a region (Rossi, 2000). We may distinguish between meteorological, agricultural, hydrological and operational drought. While the meteorological drought is identified on the basis of a deficit of precipitation, the agricultural drought depends on the soil moisture deficit which is dependent on the precipitation regime and weather, the soil characteristics and the evapotranspiration rate. The persistence of agricultural drought condition produces negative effects both on natural vegetation and agriculture. Drought periods have an important impact on water supply system causing water shortage, negatively affecting the economic and social system.

The SMD (Soil Moistures Deficit) agricultural drought index, a variation of the approach proposed by Narasimhan 2002 [11], has been calculated on a monthly basis. For the given month the index expresses the ratio between the anomaly of the monthly value compared to the average multi-annual data, and the difference between the maximum and minimum values for the entire time series available (in our case 1995-2008).

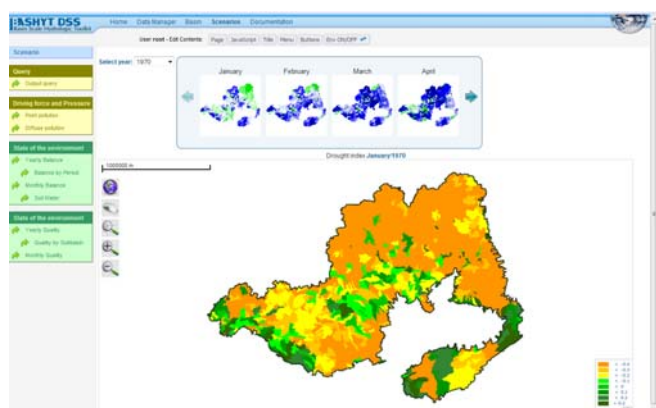


Fig. 5. Soil water deficit SMD calculate on the basis of the SWAT hydrological model.

The correct characterization of the spatial and temporal distribution of rainfall, evapotranspiration, water yield, soil water contents and the fate of nutrients and sediments is achieved using the SWAT hydrological model. By means of the web environment, the complex dynamic of surface and ground water resources are represented and can be used to design their sustainable use.

The Black Sea domain (eastern Europe), about 2 million square Kilometer basin, is analyzed by the web system that access in real time swat simulations to produce dynamic view of states of the environmental variable analyzed.

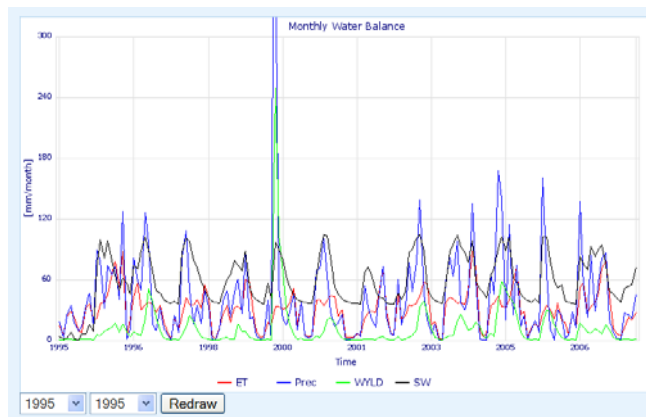


Fig. 6. Example of a (time series) graph. In details, Precipitation, real EvapoTranspiration, Water Yield and Soil Water contents estimated for the area under investigation are shown.

In figure 7, we show results of the SWAT and GETM on the web environment for the Orosei gulf (middle of the Mediterranean sea, Sardinia, Italy).

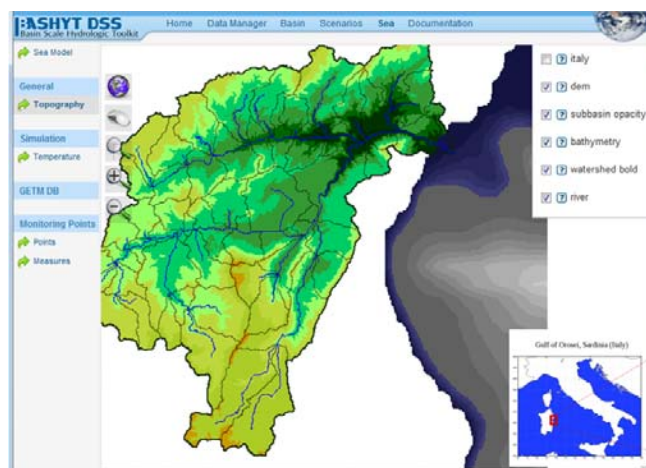


Fig. 7. Integration of the SWAT and GETM model on the BASHYT web framework (Mediterranean Basin, Sardinia, Italy).

IV. CONCLUSIONS

The SQLite architecture does not impose restrictions on distribution, size or number of files. Furthermore every single SQLite binary is statically compiled, independent from operating system and can be moved within any distributed environment to handle write, update or db management operation.

The BASHYT provides a framework for analyzing management scenarios based on valuable data and computing resources over the web. The system is based on a client/server architecture and can be used within the Internet/Intranet cyberspace, offering to the community services to extract meaningful information about the environment. In general, the web interoperability is of paramount importance to control the redundancy of replicated datasets, and it allows the user to retrieve updated certified information, avoiding the latency due to administrative and technological barriers.

As a matter of fact, environmental analysis will benefit from real-time data processing, making territorial management and planning more efficient. The software system can deeply contribute to the development and the

exchange of information relative to the environment, offering the administration standardized procedures to manage valuable data. The use of the development framework offers an infrastructure for optimizing data-sharing and solving application development problems in a multi-user environment.

The current version of BASHYT can be easily applied to many common situations in the majority of real applications, as demonstrated for the EnviroGRIDS and MOMAR projects where various case studies, located within the Mediterranean and the Black Sea basins were studied. Future work will be devoted to farther improve the reporting tools and to develop new applications.

Water protection agencies need to have efficient and reliable scientific tools to analyze complex phenomena of interest. BASHYT can represent an important contribution in the field of environmental reporting systems. Such web-based environment is designed to meet the needs of administrations involved in integrating environmental reporting procedures (based primarily on GIS, tables, graphs) and analysis tools.

ACKNOWLEDGMENT

Authors thankfully acknowledge the support of MOMAR, NUVOLA, EnviroGRIDS projects and the support of the Regione Autonoma della Sardegna (RAS).

REFERENCES

- [1] Berners-Lee, T.: Weaving the Web – The Past, Present and Future of the World Wide Web by its Inventor. Texere (2000)
- [2] Berners-Lee, T., Hall, W., Hendler, J.A.: A Framework for Web Science (Foundations and Trends(R) in Web Science). Now Publishers Inc., (2006)
- [3] Burbeck, S.: Applications Programming in Smalltalk-80: How to use Model–View–Controller (1987)
- [4] Reenskaug, T.: Models, views, controllers. Tech. rep., Xerox PARC (1979)
- [5] D. Gorgan, P. Cau, K. Charvat, D. Rodila, V. Bacu, A. Jonoski, A. Van Griensven, P. Horak, K. Abbaspour, S. Manca, G. Giuliani, N. Ray, A. Lehmann, 2010. Requirements and specifications for the development of BSC-OS Portal. Technical report of enviroGRIDS_D61 enviroGRIDS – FP7 European project.
- [6] A. Vargiu, E. Peneva, S. Manca, M. G. Mulas, F. Murgia, M. Pintus, C. Soru, R. Biella, P. Cau, 2010. A web based interface for coastal zones modelling: a test case for the Orosei Gulf in Sardinia (Italy). Proceedings of the Physics and Estuaries and Coastal Seas (PECS) Conference, Sri Lanka – Colombo, 09-2010.
- [7] H.Burchard, K.Bolding, L.Umlauf. GETM, source code and test case documentation. Version pre 1.8, <http://getm.eu>.
- [8] L.Umlauf, H.Burchard, K.Bolding. GOTM, sourcecode and test case documentation. Version 4.0. , <http://www.gotm.net>.
- [9] S.L. Neitsch, J.G. Arnold, J.R. Kiniry, R. Srinivasan, J.R. Williams, "Soil and Water Assessment Tool, User's Manual". Published 2002 by Texas Water Resources Institute, College Station, Texas
- [10] Manca, S., Cau, P., Bonomi, E. & Mazzella, A. (2006) The Datacrossing DSS: a data-GRID based decision support system for groundwater management. In: *Second IEEE International Conference on e-Science* (Amsterdam, December 4–6).
- [11] Narasimhan, B., and R. Srinivasan. 2002. Development of a Soil Moisture Index for Agricultural Drought Monitoring Using a Hydrologic Model (SWAT), GIS and Remote Sensing. Texas Water Monitoring Congress. September 9-11, 2002. Austin, TX.