

Data Analysis Features of the GESTBARRAGENS System

Alberto Silva¹, Helena Galhardas¹, José Barateiro¹, Hugo Matos¹, Jorge Gonçalves¹, Eliane Portela²

¹INESC-ID, ²LNEC

Rua Alves Redol, n° 9 –1000-029 Lisboa, Portugal
alberto.silva@inesc-id.pt

Abstract. This paper presents the technical decisions taken during the design and development of GESTBARRAGENS, which is an Information System that manages data concerning dam safety. We focus the discussion around its data analysis features, namely: (1) the generation of dynamic forms in ASP.NET; (2) the reporting mechanism implemented with MS Reporting Services; and (3) the spatial analysis based on a GIS infrastructure using the ESRI arcIMS tools.

Keywords: Safety of Dams, Data Analysis, Data Reporting, GIS

1 Introduction

According to the Portuguese Dam Safety Legislation [1], LNEC (*Laboratório Nacional de Engenharia Civil*) is responsible to keep an electronic archive of the data collected by the dam monitoring systems and for the exploitation of the information in order to maintain an updated knowledge about the dam behavior. Data is manually or automatically collected by instruments (for instance, a plumbline) located in strategic points of the dam. Then, it must be stored and handled in order to produce results that are required to support decisions concerning the dam safety.

Up to now, in Portugal, the management of these data was carried out using the so called information system SIOBE (*Sistema de Informação para Observação de Barragens de Betão*) [7], which was written in the 70's using the Fortran language, and stores data in binary and ASCII files. Through the years, new requirements concerning the type and variety of information stored as well as the data management and visualization functionalities have arisen. In order to overcome SIOBE limitations, GESTBARRAGENS was proposed in 2003 and developed along these last two years. GESTBARRAGENS is an integrated Information System (IS) that manages the data required to support the dam safety control activity [1,14,15,16,17].

GESTBARRAGENS provides several features, some of them with innovative solutions, namely: (1) Instrumentation: integrates new observation instruments, supports the dynamic management of new instruments, and manages metadata about instruments; (2) Types of observations: manages geodetic data information, information concerning visual inspections, and data provided by the automatic monitoring systems; (3) Data visualization and exploitation: accesses data through a set of reports designed to support the required types of data analyses, and spatially depicts data using a set of graphics and diagrams; and (4) Synchronization facility:

supports the deployment of the system in one or more locations (for example, LNEC and a dam owner) and the corresponding synchronization of data.

The GESTBARRAGENS system is the result of the close collaboration of three Portuguese institutions (LNEC, INESC-ID and EDP) and is currently deployed in the organizational context of LNEC and EDP.

1.1 Data analysis and visualization requirements

During the first part of the project, we took some decisions regarding the underlying technological infrastructure and the development environment. Since the system had to be a Web-based application, the following two alternatives were available: (i) a J2EE open-source solution, based on an Apache Web server and JSP, or (ii) a Microsoft .NET solution. We chose the Microsoft proprietary solution because of its higher productivity in terms of development time which was an important issue given the tight schedule and the set of requirements we had to cope with.

The nature of the system requirements in terms of data analysis and visualization was heterogeneous. In fact, several exporting formats were demanded for distinct types of data, both graphical and non-graphical. This kind of need led us to the choice of a reporting tool. We selected MS-SQL Reporting Services, in spite of its immaturity (at the time we took the decision) due to its strong integration with .NET technologies, the number of provided output formats (e.g., HTML, Excel, PDF) and its ability to support the dynamic and interactive definition of reports.

Additional requirements involved the visualization of graphical designs of dams (or its specific parts) based on 2D layouts. On the top of these 2D layouts, users should be able to select and visualize different layers, individually or in composition. Each layer is associated to a specific type of observation instrument. On the top of these layouts, users need to produce graphics that depict the temporal evolution of results. To support these requirements, three alternatives were considered: (i) to use a 2D design toolkit; (ii) to use and extend a CAD tool; or (iii) to use and extend a GIS tool. We chose the third solution, in particular the ESRI arcIMS and arcGIS modules, once again due to our tight schedule and to the flexibility provided.

1.2. Main contributions

This paper discusses the problems and solutions regarding the dam data analysis features provided by the GESTBARRAGENS system. It stresses the following aspects: (1) the generation of dynamic forms in ASP.NET, namely the generation of Web forms with labels and text fields, with repeaters, and with data grids; (2) reporting, in particular the dynamic report definition, based on a wizard approach; and (3) the GIS (Geographic Information System) based analysis, supporting spatial and spatial-temporal data analysis.

The outline of the paper is as follows. In Section 2, we briefly present the modules that compose the GESTBARRAGENS system and establish a comparison with similar systems from the point of view of their data analysis features. Section 3 describes the dynamic generation of forms. In Section 4, we detail the implementation decisions of the reporting mechanism and Section 5 the GIS features. Finally, we conclude in Section 6.

2 Overview of GESTBARRAGENS and related systems

GESTBARRAGENS is a modular system, composed by the following applicational modules: (1) gB-Observations: stores, handles and exploits data concerning the dam monitoring system; (2) gB-Visual Inspections: stores, handles and exploits data resulting from periodic inspections where anomalies in the dam are registered; (3) gB-Documents: stores, handles and exploits data about documents that concern the dam (e.g., reports, photos, diagrams, drawings); (4) gB-GIS: integrates geo-referenced information for graphical visualization; and (5) gB-Models: integrates data concerning the mathematical and physical models developed for dams. GESTBARRAGENS is a Web-based system developed on top of ASP.NET, the underlying data is stored and managed in Oracle 9i, reports are designed and implemented using Microsoft Reporting Services, and data is graphically and spatially visualized using the ESRI technology.

Several Information Systems [11,12,13,14,15] have been developed through the last two decades aiming at handling and storing data concerning dam safety. However, none of them, as far as the authors are aware of, provides data analysis features as rich as those supported by GESTBARRAGENS. Inspector (standing for Intelligent Dam Security and Control System) [13] is an IS designed and developed by CEMIG (*Companhia Energética de Minas Gerais*) and composed by fourteen modules with distinct functionalities. It is similar to GESTBARRAGENS with respect to the storage of readings and the computation of results, as well as in what concerns the management of documents and information regarding visual inspections. However, its component of data exploitation is limited to inspection reports and spatial analysis through georeferenced information.

The DamSafe IS [11] is one of the components of a decision support system developed in the context of a NATO project whose goal was the development of a research project on Dam-Break Flood Risk Management in Portugal. DamSafe encloses two databases: an alphanumeric database in Oracle to store the information about the main characteristics of the dams to be managed; and an ARC/INFO database that manages the important geo-referenced data on the flood-prone area. In terms of data visualization, it only supports data displaying using MS Excel. Finally, the main concerns of the system described in [12] were the support of real time interpretation systems connected to data acquisition units and intelligent databases to handle the off-line management of information and its interpretation.

3 Dynamic forms

Dam safety data concerning instruments, which are located in strategic points of the dam physical structure, involves three types of data: (i) information about instruments, for instance, its identification and constants of calibration; (ii) readings from different instruments, such as a plumbline reading, which comprises four distinctive readings (line radial, line tangential, cone radial and cone tangential) and (iii) the physical quantities, which are the results that are obtained from the above mentioned readings. Data is collected either manually or automatically. The latter do not require human intervention, and is here designated as Automatic Monitoring

Instruments (AMI). GESTBARRAGENS is able to manage and explore information from both systems, i. e., manual or automatic.

Currently, GESTBARRAGENS supports thirty one manual instrument types and twenty five AMI types, namely it supports the management of instrument instances, the automatic and manual storage of readings and the calculation of numeric results.

The information required for different types of instruments is heterogeneous. One possible solution to handle this heterogeneity is to design an *aspx* page with predefined forms to manage the instruments instances, readings and numeric results. However, this solution involves a huge amount of work to manually design and handle events of more than 150 different forms.

The GESTBARRAGENS system should be able to dynamically include new instrument types. A new instrument type should have different instrument definitions, readings and results. The previous approach is not powerful enough to face this problem as it is impossible to foresee the characteristics of new instrument types. Consequently, predefined and static forms cannot be used.

The alternative solution consists of dynamically generate forms. This solution involves a preprocessing step to generate new forms. In GESTBARRAGENS, we propose the following three distinct solutions based in metadata to dynamically build forms. (i) generation of forms with labels and text fields for instruments (detailed in Section 3.1), (ii) with repeaters for data readings and results (explained in Section 3.2) and (iii) with data grids for automatic monitoring systems (described in Section 3.3).

The heterogeneity of instrument definitions engages significant work to produce database objects (e.g. tables, triggers, stored procedures) and metadata information for distinct instrument types. In addition, the fact that new instrument types can be added to the system leads to the implementation of a new generic interface (wizard-type) to support and assist the insertion of new instrument types. This wizard allows the user to specify table names, column names and types. It generates an *sql script* to create database tables, stored procedures and also updates the metadata information required to dynamically define instruments.

3.1 Generation of forms with labels and text fields (instruments)

Each instrument type is characterized by a specific set of attributes. As a consequence, the system should provide different forms to edit and create new instances of distinct instrument types.

The idea underlying this solution is to define a set of pairs (*name*, *value*) for each instrument type, where the name corresponds to an attribute name or description, and the value represents the attribute value. *Labels* are used to show attribute names. *Textboxes* represent attribute values, because they allow user data entry and modifications. It is essential to provide validation mechanisms to guarantee the correction of data entered by the user.

The information required to produce forms to enter information about new instruments is called the *instrument form metadata*. It is stored in the GESTBARRAGENS database. It is crucial to map and represent this metadata in a .NET structure. Instead of creating new structures and classes to represent the instrument form metadata, we have used .NET *DataTables*. *DataTables* can store attribute names (as columns and its data types, allowing nullable values and attribute

descriptions. It also stores current database attribute values to allow the edition of instrument attributes..

At this point, we have mapped all the information needed to automatically generate new forms into our development environment,. In order to produce an interface with dynamic forms, a new generic function has been developed. It receives as arguments a *Panel* from any *Page* object where the new form will be instantiated and a *DataTable* with the specific instrument information. The function produces an HTML table.. For each *DataTable* iteration, a new line is added to the HTML table. Each line has a label with the attribute identification and a textbox to handle attribute values.

At submission time, the process is very similar. A new *DataTable* iteration is performed and textbox values are copied into the *DataTable*. Finally, a last iteration is executed to invoke the corresponding instrument stored procedure and commit modifications into database, thus making them persistent. Note that each line of the *DataTable* corresponds to a new stored procedure argument.

3.2 Generation of forms with repeaters (data readings/results)

Each GESTBARRAGENS instrument aggregates a set of data readings that produces results (by applying predefined algorithms) which are essential for dam safety analysis. Once again, a generic and dynamic interface capable to recognize and present different form fields from different instrument types is required. This scenario is different from the previous one as we can manage simultaneously all instrument data readings from a specific dam. In other words, we should be able to manage different data structures (from different instrument types) within the same form. Therefore, the solution presented in Section 3.1, where HTML tables are generated dynamically, is not powerful enough to deal with this heterogeneity in a single form.

The *Repeater* control of .NET framework has been chosen to implement this type or dynamic forms. In general, repeaters are used to iterate over some previously defined data structure, and then selecting the desired attributes to build the user interface. In GESTBARRAGENS, this data structure is not known apriori because it depends of the instrument type. Therefore, a template is required and the .NET class *ITemplate* is used to construct repeaters at execution time.

For each Repeater iteration, a new form is created to handle the instances of each instrument type. This process is similar to the generation of forms with labels and text fields presented before. In this solution, the generic *DataTable* is more complex since it includes the information of all instrument data readings. At submission time, a new Repeater iteration is performed and thus, in spite of dealing with all instruments simultaneously, they are handled individually. Data field values are validated and used as arguments in stored procedure calls to update the database. GESTBARRAGENS stored procedures store data readings in the database, and also calculate results, which are also stored in database. If results are not semantically valid, an error message is shown and is associated with the corresponding instrument data reading.

3.3 Generation of forms with data grids (AMIs)

GESTBARRAGENS supports automatic monitoring instruments (AMI). AMI data readings do not require human interaction. Data captured by AMI corresponds to instrument results. Therefore, AMI only requires mechanisms to manage the information about instruments and the data captured (results) without data

calculations. Analogously to manual instruments, the generation of forms with labels and text fields has been adopted in order to manage the information regarding AMI instruments.

A new application has been developed to send and store AMI results in GESTBARRAGENS. It is a Windows application that resides in a dam PC and communicates with GESTBARRAGENS via Web Services. Since this new application writes AMI results, a new interface has been developed to explore new data with filter mechanisms. Notice that GESTBARRAGENS also supports new AMI instrument types.

As the goal of this interface is to present results (it does not require the edition of results), the solution that uses repeaters as presented in Section 3.2, could be adopted by disabling edition functionalities. Since AMI generates huge amounts of information, this interface must present data in a very compact and structured way.

We concluded that *Datagrid* controls are better than *repeaters* to present information in such a compact way. In order to map AMI results specific attributes, data columns are added to a *Datagrid* at execution time. Since all AMI results have a common part (e.g. data capture date/time), this common information is previously associated to *Datagrid*. At execution time, and making use of metadata information, dynamic attributes are associated to *Datagrid* with a *TemplateColumn*, generating the final interface.

4 Reporting

The dam safety activity deals with a large amount of heterogeneous information. As a consequence, it is crucial to provide analysis and reporting mechanisms to visualize detailed and parameterized information in graphical or tabular format. The reporting tool adopted by GESTBARRAGENS was Microsoft SQL Server 2000 Reporting Services. The problems and solutions that emerge from the use and integration of this tool are discussed in this section.

First, as stated in section 2, since the definition of new instruments can be added, the reports that support the visualization of the corresponding information should also be dynamic. Second, it is of major importance to allow user-defined reports for dam data analysis. For example, to perform a specialized analysis, the user should be able to configure the number of graphics shown. She should be able to merge different readings or different results of distinct instrument types in a common graphic to enable the comparison of temporal evolutions. The solution adopted involves the generation of Report Definition Language (RDL) definitions at run-time. RDL is a proprietary Microsoft XML language for defining reports.

An additional GESTBARRAGENS requirement is to have a single tabular report to present all data readings and all numeric results of a specific dam at a predefined date. Since the data provided by different instruments (data readings and numeric results) obeys to a different schema, a single table or single report definition cannot be used. Hence, we use subreport mechanisms so that, different instruments have specific subreport definitions. Notice that, in order to support new instrument types, we also provide a mechanism that uses instrument metadata, and automatically generates column headers for a predefined default subreport.

4.1 Automatic RDL generation

The analysis of data temporal evolution is crucial to make decisions. For instance, a report that presents temporal evolution of results derived from different types of instruments is essential to detect dam structural problems, or an instrument anomaly.

For a specific dam, there is no way to predict the set of suitable reports. Hence, it is essential to allow users to create and configure their own reports. For instance, they should be able to define which instruments, data readings or results are included. Furthermore, we can imagine thousands of possible reports combining different data readings, results of different instruments and different instrument types. The solution proposed for automatically defining reports is based in dynamic RDL generation. Microsoft SQL Server 2000 Reporting Services does not provide other native mechanism to handle this kind of dynamism. This kind of dynamically designed reports involves three steps: (i) the definition of metadata; (ii) RDL generation and (iii) the execution of parameterized reports.

4.2 Definition of Metadata

The metadata used in reports metadata consists of information required to generate new specific reports. It includes layout and content information, such as the definition of graphics, instruments, data readings and results.

Due to the expertise required for designing this kind of reports, a simple 5-steps wizard was developed to support the definition of new:

1. **Report initialization:** to specify the names of the report and the folder where it should be stored.
2. **Specification of graphics:** to specify the number and parameters of graphics. Each graphic has a title, a scale and the option to show the information in a tabular format.
3. **Association of instruments to the report:** selection of instruments from a specific dam instrumentation list; this selection can include different types of instruments and corresponds to the instruments included in this report definition.
4. **Association between instrument data readings/numeric results and graphics:** Configure graphics defined at step 2 with data readings or numeric results of instruments selected at step 3. Notice that a single graphic can compare metrics of distinct instrument types.
5. **Report generation:** This final step checks for possible errors and generates the report.

4.3 RDL Generation

When the metadata required to automatically define reports is entirely specified, the corresponding report RDL file that will be published and later executed is created. The RDL specification is a XML file that obeys to a predefined XML Schema (XSD) delivered by Microsoft. In order to correctly generate the RDL files with new report definitions, we have implemented a set of serializable classes that automatically generate RDLs. An example of information that is serialized into RDL is dataset components.

Dataset components are parameterized with SQL queries to access data. The main difficulty occurs in creation of dataset since a single graphic encompasses different

types of instrument measures. Thus, the query is complex since the query result should be uniform, even though it accesses different data of distinct tables. Figure 1 shows an example of a query generated programmatically. It includes two results of distinct types of instruments: temperatures and volumes of drains and resistance extensometers.

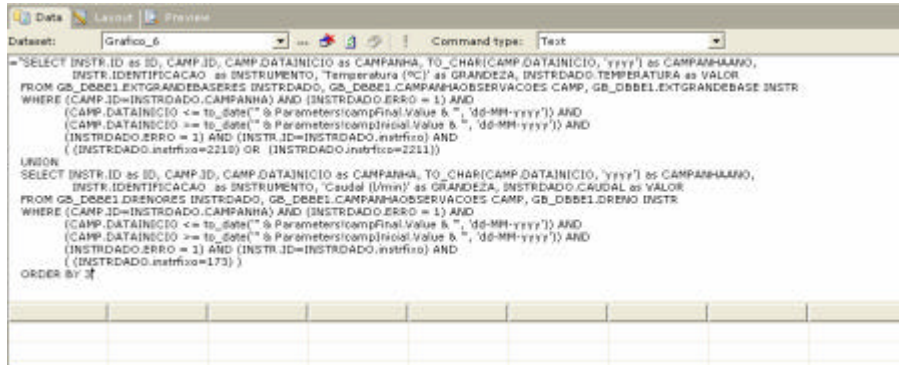


Figure 1: Generated query

Graphics creation

At this point, graphics previously defined can be created. First, an association between the datasets defined at the previous step, and the corresponding graphic is performed.

Further, dataset columns have to be mapped into graphic axes. In GEST BARRAGENS, the column *Value* corresponds to yy axe. The *xx* axe represents time (an interval between two distinct dates).

5 GIS Features

In dam safety activity, some important requirements involve data analysis with spatial visualizations of the dam structure. Furthermore, on the top of dam the structure, it is desirable to show different layers that represent specific types of instruments. Moreover, graphical representations of data captured by dams' instrumentation, and also numeric results are required to provide easy and high-level views to better understand dam spatial behavior.

ESRI products had been selected to support this kind of data visualization.

In order to fully integrate GEST BARRAGENS with the GIS module, we had to assure that the GIS component runs under Microsoft IIS sever. Thus, the application was developed in .NET technology, in C# and JavaScript languages. Web pages were developed according to ASP.NET technologies. All *http* requests were redirected to Microsoft IIS server, which is responsible to communicate with the ArcIMS Web-based map server.

The Web interface provides mechanisms that allow users to easily navigate and explore physical elements of dams. They are able to monitor mouse click events, for

instance to select a dam or an instrument type; or activate common GIS features such as zoom, pan, undo, redo, measures, printing, spatial objects identification and information.

Graphical representations of data on the top of physical dam structure, such as diagrams or graphics are the utmost features of our GIS module. Figure 2 shows examples of spatial graphics delivered by GEST BARRAGENS GIS module.

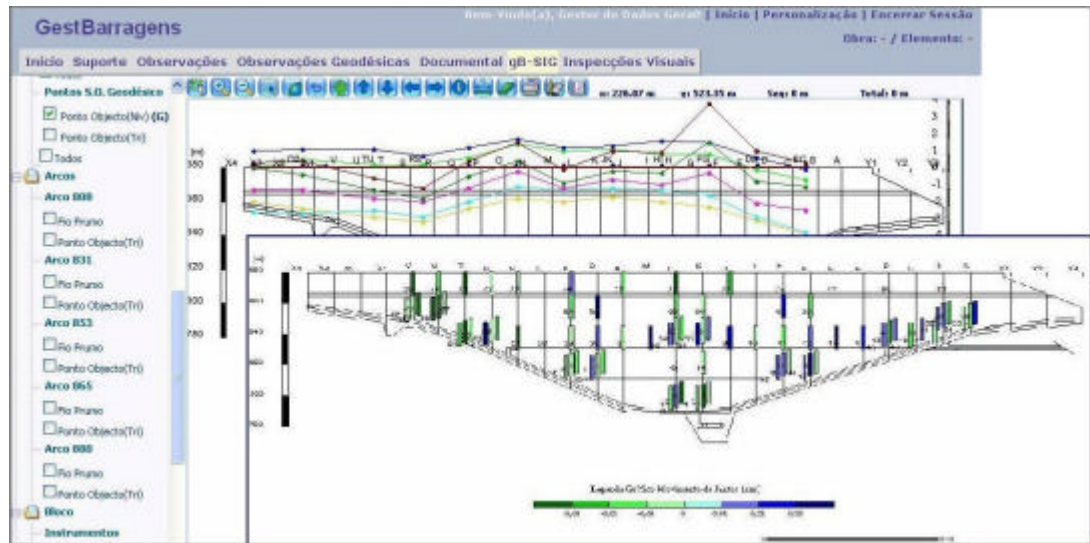


Figure 2: GESTBARRAGENS spatial analysis

There are some complex algorithms that compute data stored in the database and produce metrics presented graphically, as suggested in Figure 2. These kinds of complex visualizations provide a spatial-temporal data analysis capabilities, and are the most powerful but complex data analysis.

6 Conclusions

GESTBARRAGENS is an information system that manages data concerning dam safety. It is a modular and multi-user Web-based system that supports complex requirements. When compared with international scenarios, such as [14, 15], it is an innovative and flexible system. In this paper we stress the GESTBARRAGENS data analysis features. In particular, we discuss the decisions taken and the involved trade-offs, and then explain some design/implementation details.

The analysis and selection of the right tools and technologies, such as ASP.NET, Reporting Services and arcGIS, was a key contribute for the success of the project.

We consider that the problems and solutions discussed in this paper, for the dam safety domain, can be adopted with benefits to other areas of civil engineering, such as the monitoring of the behaviour of bridges, roads or other buildings.

References

- [1] RSB, *Regulamento de Segurança de Barragens, Decreto-lei N.11/90 de 6 de Janeiro*, Diário da República, Lisboa. 1990.
- [2] Portela, E., *Novas Metodologias de Apoio ao Controlo de Segurança de Barragens de Betão*, Tese de Doutoramento, IST, Outubro 1999.
- [3] Schmidt, T., Grounds, M., *Dam Safety Management Tools (DSPMT), 21st USSD Annual Meeting and Lecture – The Future of Dams and Their Reservoirs*, Denver, Colorado, 2001.
- [4] Silva, A. R., Galhardas, H., Portela, E., *Relatório Técnico INESC-ID, “gestBarragens: Sistema Integrado de Gestão da Informação para o Controlo de Segurança de Barragens”*, Versão 1.3, Fevereiro de 2002.
- [5] Silva, A. R., Galhardas, H., Portela, E., *Relatório Técnico INESC-ID, “O Modelo Multi-institância Federativo do GESTBARRAGENS”*, Julho de 2005a.
- [6] Silva, A. R., Galhardas, H., Barateiro, J., Portela, E., *Relatório Técnico INESC-ID, “Especificação de Requisitos Técnicos do gestBarragens”*, Versão 2.0, Dezembro de 2005.
- [7] Silva, H., Amante, J., Tavares de Castro, A., *Sistema de Informação para Observação de Barragens de Betão (SIOBE), Manual de Utilização*, LNEC, 1993.
- [8] Soares, H. e Custódio, M., *gB-Sistema de Informação Geográfico, Relatório do Trabalho Final de Curso*, LEIC, IST/UTL, Julho de 2005.
- [9] Martins, T., *gB-Documental e gB-Inspeções-Visuais, Relatório do Trabalho Final de Curso*, LEIC, IST/UTL, Setembro de 2005.
- [10] Joseph O’Rourke, *Computational Geometry in C*, Cambridge University Press, 1993.
- [11] Fernandes, J.P., Andrade M.J., *A database for dam safety management*, Dams and Safety Management at Downstream Valleys, 1997 Balkema, Rotterdam.
- [12] Lazzari M, Salvaneschi P., *Improved Monitoring and Surveillance through Integration of Artificial Intelligence and Information Management Systems*, 10th Conference on Artificial Intelligence for Applications, San Antonio, Texas, March 1994.
- [13] Balbi D, Fusaro T., *INSPETOR – Sistema Inteligente de Controle e Segurança de Barragens*, Comité Brasileiro de Barragens, XXV Seminário Nacional de Grandes Barragens, Salvador, Brasil, October 2003 (in portuguese).
- [14] ASDSO, Association of State Dam Safety Officials, <http://www.damsafety.org/>
- [15] Dam Safety Program Management Tools, <http://www.safedams.org/>
- [16] Comissão Nacional Portuguesa das Grandes Barragens, <http://cnpgeb.inag.pt/>
- [17] Seminário “Barragens: Tecnologia, Segurança e Interação com a Sociedade”, de 27 a 29 de Outubro de 2005, CNPGB, 2005