

Component-based Architecture for e-Gov Web Systems Development

Camilo Carromeu, Débora Maria Barroso Paiva,
Maria Istela Cagnin, Hana Karina Salles
Rubinsztein, Marcelo Augusto Santos Turine
Faculdade de Computação
Universidade Federal de Mato Grosso do Sul
Campo Grande, Brazil
{camilo, debora, istela, hana, turine}@facom.ufms.br

Abstract—In spite of the wide coverage of Internet and the need for Web systems to support various governmental tasks, the Brazilian state of Mato Grosso do Sul (Central West region) did not have a technological infrastructure which was sufficient to meet these needs in 2001. This led to setting up the Laboratory of Software Engineering (LEDES) at the Department of Computing and Statistics (DCT), the current Computing Faculty (FACOM), at the Federal University of Mato Grosso do Sul (UFMS), which is now collaborating effectively with the state government providing technological solutions, especially in Web sites and Web applications (WebApps). Developing these solutions has made it possible to capture patterns, define flexible architecture and subsequently set up a process of a Software Product Line (SPL) to develop WebApps in e-gov domain, as well as create computational support tools which automatize this process. This paper presents such computational support tools, lessons learnt during the most relevant projects which have been developed since the creation of LEDES, and the architecture for e-gov web systems development, resultant from early experience.

Keywords-e-gov; software architecture; software product line

I. INTRODUCTION

The emergence and growth of Internet from the 1990s caused significant impact on the way that tasks are carried out in various fields (business, industry, government and education, among others).

However, in 2001, there was still a need both in Web systems [1] to support activities from the Brazilian state government of Mato Grosso do Sul, as well as Web sites to publicize governmental content whose access is an interest and right of any citizen. It should be mentioned that this reality was not very different in relation to the other states of Brazil. The same problem arose in various departments at the Federal University of Mato Grosso do Sul (UFMS) because employees were doing tasks manually and, therefore, adding to the number of errors and inconsistent information due to the fact that the same information could be registered in more than one department.

This led to the setting up of the Laboratory of Software Engineering (LEDES) at the Federal University of Mato Grosso do Sul (UFMS), which has started to develop software solutions for the state government, the university, and subsequently the federal government.

Karin Breitman
Departamento de Informática
PUC-RIO
Rio de Janeiro, Brazil
karin@inf.puc-rio.br

The various solutions which have been developed made it possible to train qualified staff in the Brazilian Central West region, improving the skills of the Software Engineering group to develop Web solutions, mainly e-gov, and forming a technological infrastructure to enable the transference of knowledge from universities to the government.

Drawing from the various experiences of the team, a flexible architecture to develop WebApps by reusing components [2,3] could be defined, which made up the basic core of any WebApp. To facilitate the efficient use of this architecture, support tools were developed such as the Titan Framework [6].

The aim of this paper is to present an architecture to support the development of WebApps, obtained from various projects developed by LEDES regarding the development of WebApps, especially in e-gov domain.

The paper is organized in five sections: related work is presented in Section 2; the main projects which were developed in LEDES, as well as lessons learnt are reported in Section 3; the component-oriented architecture to develop WebApps and the support tools are presented in Section 4; and finally in Section 5, conclusions and suggestions for future work are discussed.

II. RELATED WORK

Many works related to e-gov domain may be discussed. For example, the SIGAP (Integrated System of Managing Groups and Research for the state of Pará) is a software solution developed for e-gov domain. The aim is to collect and manage information concerning researchers, laboratories, and current research groups in the Brazilian state of Pará. Moreover, the system makes it possible to produce reports and useful indicators for the governmental research agencies of the state [2].

Fig. 1 shows the flow of information from SIGAP presenting the integration of information sources related to scientific research (Academic Curriculum Vitae and the CNPq (The National Council for Scientific and Technological Development) Research Group Directory). Additionally, Fig. 1 indicates the supply of data from the researcher concerning more specific information within his/her research context.

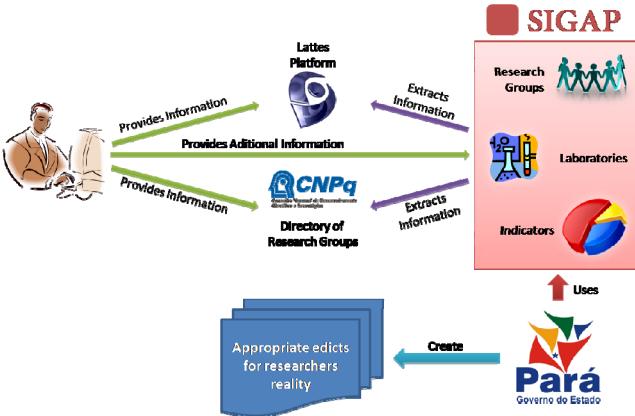


Figure 1. Flow of information in SIGAP [2]

As a result, the governmental agencies can visualize information about researchers, identify their needs, and use the indicators and reports to implement and carry out policies which are more appropriate for the reality of science, technology and innovation in the state of Pará.

Tools are also used in e-gov to manage software processes. The Information Technology and Communication Centre (CTIC) at the Federal University of Pará (UFPA) started working on improving the software processes using WebAPSEE [3] as a support tool to manage software processes. The main objective of the tool is to help model and carry out software processes. Some of the important functions of the tool are the support and reuse of processes, as well as the control of artifacts resulting from project management (schedules, cost planning, and human resources). The tool has helped the group to obtain the MPS-Br G level certificate [20], the Brazilian model to improve processes developed based on ISO/IEC 12207 standard [28] and CMMI model [29].

The Stela Group (<http://www.stela.ufsc.br/>) from the Federal University of Santa Catarina was founded in 1995 and has also developed various projects in e-gov. Among them, the following can be highlighted: the Research Group Directory in Brazil, a project from CNPq to gather together information concerning all the work of research groups in Brazil; Academic Curriculum Vitae together with the first version of the CV System, an information system base which has integrated applications used by researchers, students, and managers of science and technology to register information; and Pro-Collection systems with the aim of helping in the evaluation process of Brazilian post graduate courses.

Developing the foregoing technology made it possible to consolidate the Stela Group and, from September 2002, the group officially stopped working on the activities and its team set up the Stela Institute (<http://www.stela.org.br/>). Currently, the institute has various information platforms and knowledge for different research and teaching institutions, governmental and international agencies, such as MEC/Inep (Ministry of Education/National Institution of Studies and Anísio Teixeira Educational Research), Anvisa (National Health Surveillance Agency), OPAS (Pan-American health

Organization), Fiocruz (Oswaldo Cruz Organization), CNPq (The National Council for Scientific and Technological Development), CAPES (Brazil Higher Education Consortia Program), ANP (National Agency of Petroleum), among others.

The conceptualized architecture for the platforms of the electronic government conceived by the researchers of the Stela Institute takes into account the interests of all those involved in the domain. The application of this architecture can solve two common problems in governmental initiatives in information management: the lack of integration and the low quality of information. It is made in the shape of a pyramid, as shown in Fig. 2. The base layer, Pieces of Information, deals with setting up schemes which make the sharing of common information in future systems possible. The second layer deals with setting up repositories and capture systems, dealing with and storing information respecting the pieces of information defined in the first layer. The third layer deals with instruments to present information on the Web with search services and dynamic up-to-date information. The top layer of the pyramid contains knowledge systems projected to generate new knowledge from pieces of information, information systems, and web portals [30].

III. BACKGROUND

LEDES was founded in 2001 at UFMS with the aim of meeting an increasing demand from the state government and the university itself to develop software for management and taking decisions. On this occasion, the first study group in Software Engineering at UFMS was set up, which was the laboratory where practical experiments could be done with the students.

The development of the first projects at LEDES was carried out in an ad-hoc way, in other words, not following the standardization and development process.

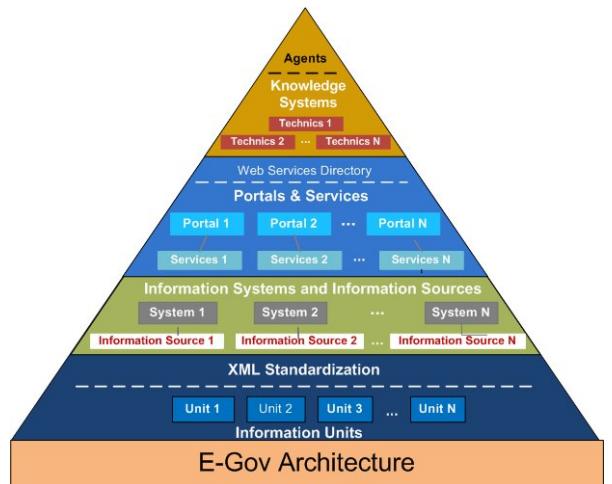


Figure 2. Conceptualized Architecture for e-gov platforms

The planning and the choice of technology and tools to support the development of the projects was solely based on experience far from other universities, and often in completely different real-life situations from UFMS. As a consequence, the technology chosen was based on proprietary software, as was the case of DBMS (Database Management System) SQLServer, and ASP and Delphi languages (procedural programming). Project management and configuration tools were not used and documentation was not produced.

The initial development team was basically made up of undergraduate students who were learning to develop software and did not have experience in Software Engineering. The first project of the laboratory was SIGS (Information System and Social Management) [21] for social management programs. This software already had a stable version, arising from a project developed at PUC-SP (another Brazilian university) and was extended having new functions.

The first projects initiated and completely developed by LEDES were Online Fapec (<http://www.fapec.org>) and the web portal of the State School (<http://www.escolagov.ms.gov.br/>). Software Engineering techniques were used, although in a simple and informal way, considering the incremental development. The aim of these two institutional web portals was to meet the demand of spreading information and grouping intranet systems from these foundations. SIGS was developed in ASP and SQLServer, proprietary technology. Fapec Online and the State School web portal were developed in PHP 3 and MySQL, free technology and open code.

From these foregoing projects, there was an initiative to develop a specific system for the Support Foundation for Research in the State of Mato Grosso do Sul – Fundect (<http://www.fundect.ms.gov.br>), called Fundect Online, reusing some modules of the Fapec Online project. In this project, the largest until now, various problems in the development process could be identified, such as lack of documentation. However, other experiences were put into practice, for example, the emphasis on usability of the system. As a result, the conception and implementation of a help module accessible at any point of the system were considered. There was also progress in terms of the development process, mainly in adopting a management tool for projects called GForge (<http://gforge.org/>).

In spite of the projects mentioned having been developed in a gradual way, in general, without using Software Engineering techniques, they met the objectives initially defined in a satisfactory way. In addition, the reusability of modules helped to identify standards of the project in the systems which were implemented.

From these initial experiences in setting up the Governmental Institutional web portals, the development of a tool was started called Pantaneiro [22]. The aim was to supply a generator of Websites in the e-gov domain, created especially to meet the needs of the state government of Mato Grosso do Sul. This tool started to be used from the end of 2003 and, up until now, 56 governmental sites have been set up.

At the beginning of 2004, reusing various modules of Fundect Online, a WebApp was developed to manage extension projects at UFMS, called SIEX. Until now, university extension programs in Brazil did not have any system to automatize submission and evaluation projects. SIEX fulfilled this role well as it came up with free software available to all the other Higher Education Institutes. This resulted in the extensive use of SIEX by academic community. In next five years (2004-2009), its relevance to the community of institutions which used and supported the initiative grew, culminating in its progress for SIGProj (<http://sigproj.mec.gov.br/>). The aim was to manage teaching projects, research, extension, and student issues in the Ministry of Education – MEC and was used by academic institutions all over Brazil.

In 2006, the Fundect Online System was presented together with the National Council of State Foundations of Assistance and Research (CONFAP) as a successful case in submitting information processes and following proposals. The positive results that the system brought to Fundect motivated other research agencies to look for LEDES to set up Fundect Online for other situations. Identifying this new demand we extended the original system with the aim of meeting the common needs of all research agencies. This new project had the SIGFAP abbreviation. The first part of the system was implanted at the Foundation of Assistance and Research in the state of Pará (FAPESPA) in 2007. In the following year, SIGFAP was set up for the Foundation of Assistance and Research in the Amazon (FAPEAM) and in 2009 for the Foundation of Assistance and Research in the State of Piauí (FAPEPI).

Various other institutional management systems were set up in LEDES, most of them to meet the needs of UFMS. Most of these systems were institutional web portals for internal university departments, including management modules to help manage data and register resources and local people.

One of these systems was the Academic Online System (SISCAD), developed by LEDES and the Computing Centre (NiN) at UFMS as a centralized platform and the only one to group and control academic data. All the information concerning the academic life of each university student is centralized by means of SISCAD.

Another example is the Information System of Institutional Evaluation (SIAI), which processes information concerning the evaluation of teaching staff and technicians of UFMS. Similar to SIGAP (Section 2), SIAI imports data from different systems at UFMS and the Curriculum Vitae.

Having an increasing demand of new projects at LEDES, a search for reusing codes from modules of old systems in new ones began naturally. By doing this, various patterns which repeated themselves in all WebApps were gradually developed and recognized. As a result, the development of an architecture started (Section 4), and was implanted originating the Titan Framework. This tool began to be used in 2005 in all the new LEDES projects. The aim was to facilitate the new WebApps in the e-gov domain.

In its first version, Titan was only a CMS (Content Management System) which could be parameterized. It was

implemented in PHP 4 and MySQL avoiding having to implement various modules again, which all the WebApps contain, such as the access control and user management. However, its parameters needed knowledge of the language and the functions needed the copy-paste program code. This initial version was used to develop SISCAD. Although the Titan version was not very mature, the architecture was valid.

In 2007, a new version of Titan Framework was made available. This version contained various improvements, presenting the implementation of the developed architecture.

IV. ARCHITECTURE FOR E-GOV SYSTEMS DEVELOPMENT

Having the objective of being reusable in different levels of abstraction, software patterns [7, 8, 9] appeared in the 1990s trying to gain the experience acquired in developing software in terms of problems and solutions. As well as providing the reuse of solutions, the patterns helped to improve communication between developers, who could conduct discussions based on these patterns [9]. For those developers who know the patterns, the simple citing of their names brings a significant semantic content and, therefore, a detailed explanation of the solution is not needed.

Throughout the resolution process of a particular problem, rarely do the specialists invent a new solution which is different from the existing ones. Various solutions are known according to someone's own experience or from other professionals. Therefore, when new problems arise, it is common to recuperate and remember old solutions, thinking in pairs "problem/solution" [7, 9]. The software patterns describe process solutions for problems which often occur when developing software. The function is to recuperate and document the experience gained by developers during their professional practice. According to Fayad, Johnson and Schmidt [10], patterns describe solutions which were approved during the time of use, so that common characteristics which could be reused in new projects and at different levels of abstraction, not only codes, but also process, architecture, analysis, projects and programs, among others, could come to light.

Taking this into account, the various experiences of LEDES in developing applications in e-gov domain resulted in abstracting analysis, architecture, and codification patterns. Aiming to consolidate an agile development process for this domain, the team has chosen to use the Software Product Line (SPL) reuse technique automatized by using frameworks and application generators.

According to Chastek [23], a SPL defines a family of Software Products (SPF) in a domain which shares features. According to Gimenes and Travassos [24], a SPF defines a set of software products having similar characteristics to define a common infrastructure of structuring artifacts which make up the products and the parameterization of its differences. Therefore, from the patterns abstracted from different members of the SPF in the domain, the features model was created and the initial architecture of SPL was established.

The created SPL process is based on the PLUS approach (Product Line UML-Based Software Engineering) [25]. The

aim is to develop systems in the family of products in e-gov domain in an interactive way, compatible with the USDP (United Software Development Process), RUP (Rational Unified Process) and the Boehm's spiral process model [26] and consider a project based on components. The process that PLUS uses, called ESPLEP (Evolutionary Software Product Line Engineering Process), has two life cycles: Software Product Line Engineering and Application Engineering.

In the SPL Engineering cycle, the SPL engineer defines the similarities and the variables from the abstracted patterns and the analysis of SPF requirements in the business domain. The SPL artifacts developed during this cycle are the case model in use, feature models, analysis model, SPL architecture and reusable components. In Fig. 3, the creation of SPL engineering and the initial version of component repositories from LEDES using SPF pattern abstraction are presented.

In Application Engineering, an application is developed from the requirements of the application engineer and the SPL repository artifacts, i.e. a new member of SPF. The use case model of SPL is adapted giving origin to the use case model of the application. Therefore, having the application architecture and repository reusable components, the application can be constructed. In Fig. 4, the ESPLEP process is presented.

The application of the PLUS approach allows specifying reusable artifacts in the domain of e-gov family products and the formal process of developing SPF new members. However, in order to systematize the reusing process, implement the components, and be able to generate new WebApps, using automatized tools is essential. Taking this into account, an environment where applications can be generated was defined, which implements the SPL architecture, formed by two tools: the Titan framework and the Titan Architect application generator.

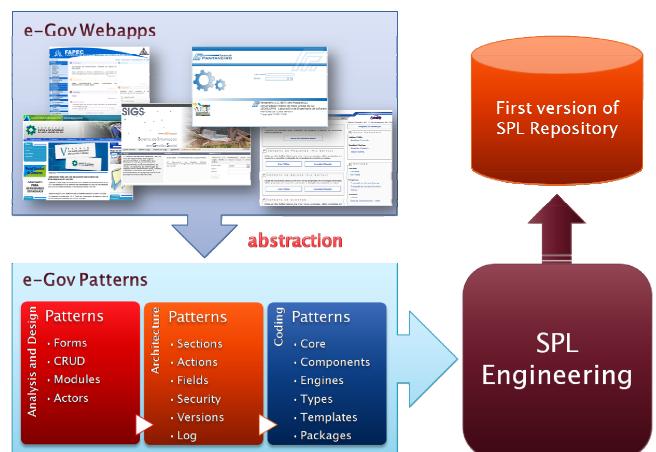


Figure 3. Abstraction of patterns in the domain and SPL Engineering specification

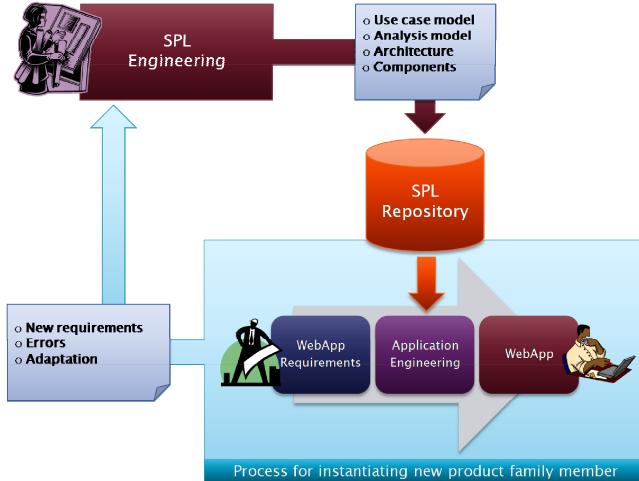


Figure 4. ESPLEP Process of the SPL for e-gov domain

Frameworks are defined as semi-complete and reusable applications which, when specialized, produce personalized applications within a specific domain [11]. Although they are typically for reuse purposes, frameworks have some characteristics, such as dependency on well-defined interfaces, reuse of the project and architecture, use of patterns, which can help in developing more organized systems with architectures to make adaptation and extension easier. The structure of a framework can be defined as a set of classes which contain the abstract project of solutions for a family of problems providing the gradual reuse of the classes [11]. It is made up of a collection of abstract and concrete classes, as well as interfaces between them, representing the project of a subsystem [12].

In summary, a framework is a large model of an application domain and consequently can be developed from a set of applications in the domain which acts as sources of information [16]. The most relevant methodology of framework development for this work was the example-oriented projects [17], where the abstraction of the domain, which is represented by the framework itself, is obtained by real cases.

Frameworks are made up of fixed parts and variable parts. The fixed parts are called frozen-spots and are stable when the framework is being used in various applications. The variable parts are called hot-spots and must be adapted in a particular way at each instance of the framework [13]. To automatize the application engineering cycle of ESPLEP, the SPL feature model, formed by similarities and variabilities was mapped out for the Titan framework specifying the frozen-spots and hot-spots.

The Titan framework architecture is made up of a core and component repository, as can be seen in Fig. 5. The core is the implementation of all the similarities of SPL and consequently has a firm characteristic, independent of configuration of the instance in the domain. It is responsible for receiving the incoming configuration files of instances (XML and SQL), as well as generating an application when being executed.

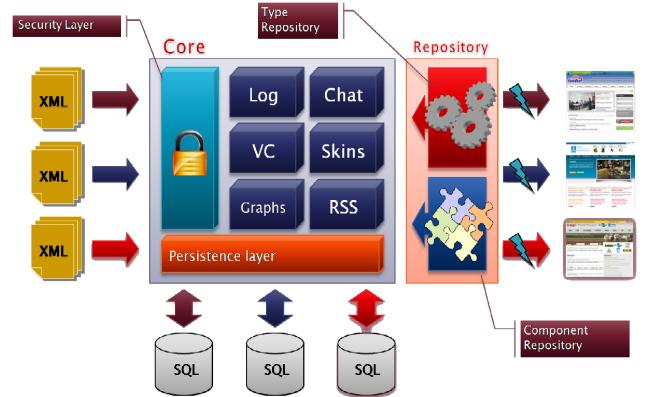


Figure 5. Titan Framework Architecture

The Titan framework implements the SPL repository in the format of a component repository which is able to reuse data types, functions, and code templates. These components are the SPL variables which can be parameterized by XML (Extensible Markup Language).

The Titan architecture has grey-box, flexible, and extendable characteristics. In other words, components can be created by its object-oriented API to meet the new requirements of the domain, correcting errors and making adaptations, thus implementing the Engineering SPL cycle.

In order to make the process of creating new applications for SPF even more agile, an application generator called Titan Architect was specified and implemented, which generates the XMLs and SQLs files necessary to instantiate the Titan Framework. According to Cleaveland [27], application generators are software systems which transform specifications into an application. The specifications describe the problem or task which has to be carried out by the manager. These specifications can be modeled in graphic form, written in an intermediate language or even created interactively, whereby the user chooses the characteristics he/she wants by selecting a sequence of forms or menus.

The Titan Architect makes configurations of new instances in the domain in an interactive way possible by four configuration steps. In the first step, main configurations and visual characteristics of applications are defined. In the second step, the application actors, their data and how their access to the system will be, are configured. In the third step, the functions of the system are configured, i.e. the parameters which will be passed to the components are defined. The last step generates the download of the new application. The first step of the Titan Architecture is shown in Fig. 6.

The Titan Framework and Architecture automatize the SPL process of generating new applications for the family of products in the e-gov domain requiring different levels of knowledge. The use of the Titan Architecture eliminates the need for programming; however the parameterization of variables is limited. The Titan Framework makes the configuration of business rules and parameters of the new application possible by editing the XML file and, if it is not sufficient, it can program new software artifacts.

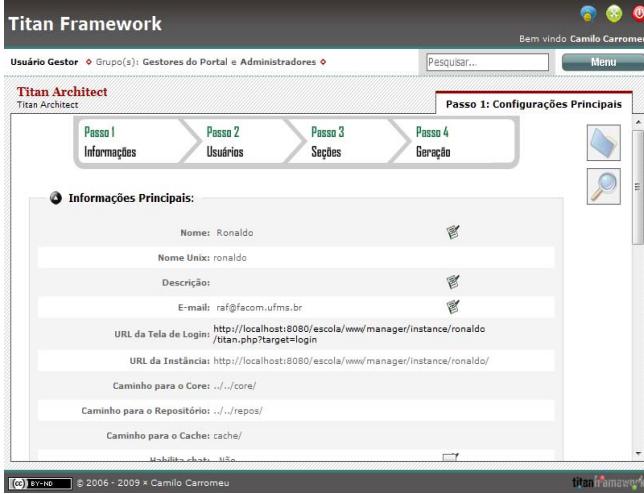


Figure 6. First step of Titan Architecture

This characteristic, from the educational point of view, is convenient to integrate students from all years in the LEDES project.

The process and the tools have been constantly validated. Since it was created, many new WebApps were developed and others are currently being developed in LEDES. All the new applications for e-gov produced in the laboratory use SPL and the cited tools.

V. CONCLUSIONS

This paper presented an architecture oriented for components to develop WebApps in e-gov domain. This architecture was conceived by identifying patterns in various projects developed in this domain by LEDES/UFMS. Using SPL Engineering, a SPL repository was created to facilitate the instances of new members for the family of software products in e-gov domain. The instances are carried out using the Titan Framework, which was created specifically to concretize the architecture which was established initially, with the support of the Titan Architecture.

As a result, the Titan Framework provides the development of various applications, with less time and costs, which meet the needs of different current public departments, such as the Web portal at the Federal University of Mato Grosso do Sul – UFMS (<http://www.ufms.br/>), the National Phone-in System of Accusations – DDN 100 (<http://ddn.ledes.net/>) for the Special Secretary of Human Rights from the Presidency of the Republic and the Information System of Evaluating Cultural Projects – SIAC (<http://minc.ledes.net/>) for the Ministry of Culture.

The architecture has been constantly developed and case studies in other domains, such as agro business have been carried out. Moreover, new implementations, for instance for a mobile platform, show its considerable flexibility. Work will also be done to improve the accessibility of the WebApps by the Titan Framework so that international standards (i.e. WCAG [18]) and the Brazilian (i.e. e-MAG [19]) can be reached and established.

ACKNOWLEDGMENT

The authors thank CNPq (process number 551017/2007-4 and Casadinho Project) and Fundect (process number 23/200.094/2008) for funding this research.

REFERENCES

- [1] A. Ginige, "Web Engineering: Managing the Complexity of Web Systems Development," Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering, New York, pp. 721-729, 2002.
- [2] A. Lemos, L. Nascimento, E. O. Sales, and C. A. L. Reis, "Using practices to generate projects and develop a support system for research networks in the state of Pará," II Workshop Software Project Management, 2009, Ouro Preto - MG. Brazilian Symposium of Software Quality, 2009. (In Portuguese)
- [3] A. Lima, A. Costa, B. França, L.C. Reis, and R. Reis, "Flexible management of software processes using WebAPSEE," 19º Brazilian Symposium of Software Engineering (SBES'06) – Tool Session. Florianópolis, SC, Brasil, 2006. (In Portuguese)
- [4] J. Cheesman, and J. Daniels, "UML components: A simple process for specifying component-based software," Addison-Wesley Publisher, 2000.
- [5] I. M. S. Gimenes, and E. H. M. Huzita, "Component-based development: Concepts and techniques," Ciéncia Moderna Publisher, 2005. (In Portuguese)
- [6] C. Carromeu, "Software product line in the process of generating Web support systems to manage projects," Master's Thesis, Federal University of Mato Grosso do Sul, 2007. (In Portuguese)
- [7] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerland, and M. Stal, "Pattern-oriented software architecture - a system of patterns," Wiley & Sons, 1996.
- [8] J. O. Coplien, "Software design patterns: Common questions and answers. The patterns handbook: techniques, strategies, and applications," New York, NY, USA: Cambridge University Press, 1998.
- [9] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, "Design patterns - elements of reusable object-oriented software," Addison-Wesley, 1995.
- [10] M. E. Fayad, D. C. Schmidt, and R. E. Johnson, "Domain-specific application frameworks," Wiley Computer Publishing, 1999.
- [11] B. Foote, and R. E. Johnson, "Designing reusable classes," Journal of Object Oriented Programming, v. 1, n. 2, pp. 22-35, 1988.
- [12] I. Sommerville, "Software Engineering," Addison-Wesley, 8th ed., 2007.
- [13] E. K. Shimabukuro, "A Generator of configurable applications," Institute of Mathematical Science and Computing - ICMC-USP, São Carlos, SP, Brazil, 2006. (In Portuguese)
- [14] M. E. Fayad, and D. C. Schmidt, "Object-oriented application frameworks," Communications of the ACM, ACM Press, New York, NY, USA, v. 40, n. 10, pp. 32-38, 1997.
- [15] A. Yassin, and M. E. Fayad, "Application frameworks: A survey," John Wiley & Sons, 1999.
- [16] R. P. Silva, "Support to develop and use frameworks and components," Federal University of Rio Grande do Sul, Porto Alegre, RS, Brazil, 2000. (In Portuguese)
- [17] R. E. Johnson, "How to design frameworks," Tutorial at OOPSLA'93, 1993.

- [18] WCAG 2.0, “The WCAG 2.0 documents,” 2008b. URL: <http://www.w3.org/WAI/intro/wcag20>.
- [19] e-MAG, “Recommendations of accessibility to construct and adapt content from the Brazilian Government on the Internet,” Department of Electronic Government, Secretary of Information Logistics and Technology, Ministry of Planning, Budgets and Management. Version 2.0, 2005b, 45 pp. (In Portuguese)
- [20] Softex, “MPS.BR – Improving the Brazilian software process – General guide,” Version 1.2, June, 2007. (In Portuguese)
- [21] M. A. S. Turine, “Information Management and Technology as support for social inclusion: The experience of the Information system in social management,” SIGS. 2003. (In Portuguese)
- [22] H. C. Sandim, “Pantaneiro: Framework of Web applications for e-gov platforms,” Master’s Thesis, Federal University of Mato Grosso do Sul, 2009. (In Portuguese)
- [23] G. Chastek, “Object technology and product lines,” OOPSLA 97: ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications (Addendum). Atlanta, Georgia, USA: ACM Press, 1997.
- [24] I. M. S. Gimenes, and G. H. Travassos, “Focus of product lines to develop software,” Event at XXII SBC Congress - SBC2002. XXI Updating in Computing seminar. Brazilian Society of Computing, 2002. (In Portuguese)
- [25] H. Gomma, “Designing software product lines with UML,” Addison-Wesley, 2005.
- [26] B. A Boehm, “Spiral model of software development and enhancement,” IEEE Computer 21(5): 61-72, 1988.
- [27] J. C. Cleaveland, “Building application generators,” IEEE Software (pp. 25-33), 1988.
- [28] ISO/IEC 12207:2008 – International Organization for Standardization / International Electrotechnical Comission. ISO/IEC 12207 Systems and Software Engineering – Software Life Cycle Processes, Geneve: ISO, 2008.
- [29] M. Phillips, M. B. Chrissis, M. Konrad, and S. Shrum, “CMMI: Improving and integrating,” PA: Software Engineering Institute, Carnegie Mellon University, 2003.
- [30] Stela Institute, “Platforms for e-government,” URL: <http://portal.stela.org.br/competencias/metodologia-egov>.