

Peony: a Web Environment to support pattern-based development

Rosana T. Vaccare Braga

Alessandra Chan

ICMC - Universidade de Sao Paulo – Av do Trabalhador Saocarlene, 400, Sao Carlos, SP, Brazil
{rtvb,alechan}@icmc.usp.br

Abstract

The development of Web applications using software patterns, test requirements, and established processes helps to increase teams productivity and the final product quality. However, there is a lack of tools for supporting developers on the use of software patterns in the several stages of a development process, beyond assisting the validation of the reused solutions. Thus, this article presents a Web environment, named Peony, that aims at facilitating the reuse of software patterns and test requirements during software development. Peony allows the software engineer to include the development processes followed by the organization, and to associate patterns to their phases and/or activities, so that during the development of a particular project the developer can more easily reuse these patterns. Also, for each pattern, Peony suggests test requirements for assisting the validation of applications under development.

1. Introduction

The expansion and popularization of the World Wide Web have encouraged the development of Web applications. There is an increasing demand for more complex Web applications, whose development must be done with quality and urgency. Several Web development methods have been created to guide the life-cycle of these applications, as well as tools supporting their use [10, 12, 5].

The use of patterns in the development of software applications is also recommended to enhance productivity and quality [8]. Patterns provide abstract solutions to recurring problems, and their reuse helps to make application development faster. In particular, Web Applications can be built more easily through the application of navigational patterns [9], business rules patterns [2], among others [6, 13]. According to Garzotto [9], the design of complex Web applications can benefit from the adoption of patterns, such as enhancing the final product quality, reducing efforts, number of developers, and development cost. In order to minimize

errors and facilitate use, tools supporting the use of patterns have been developed [16, 11]. But there is a lack of environments and tools supporting their use during all stages of applications development.

Among the main activities to ensure software quality are VV&T (Verification, Validation, and Testing) [18]. Some initiatives have been proposed in a recent work [3] associating tests to patterns, to minimize the time spent in VV&T.

This paper presents the Peony environment, which aims at supporting the use of software patterns during all development process stages. Therefore, software patterns can be previously associated with these stages, so they can be suggested to the developer during project execution. Patterns of several categories, such as design patterns, analysis patterns, human-computer interaction patterns, etc, can be included, helping developers to find solutions for problems found during applications development. Moreover, Peony offers the possibility to associate test requirements to software patterns, helping VV&T activities.

The paper is organized as follows: Section 2 discusses related work. Section 3 presents the Peony environment. Section 4 presents the final remarks and future work.

2. Related Work

Several environments have been presented to support development processes execution, most of which allow the explicit representation of a process and its related products, integrating tools for software development with process modeling tools. Examples of such environments include WebAPSEE (Web Process-Centered Software Engineering Environments) [15] and ODE (Ontology-based software Development Environment) [1]. WebAPSEE eases the modeling, execution, and maintenance of software development processes. Other functionalities are being implemented to support processes simulation, reuse and instantiation. ODE is a focus-centered software development environment, based on ontologies, and used to define a particular type of process (ODE [7]).

On the other hand, several pattern repositories are available ¹, but it is not straightforward to use them in a systematic way, because developers would need to search for the right pattern in various sources, then to decide if they fit the problem context, the particular process stage being performed, the organization goals, etc. Some organizations maintain a repository of frequently used patterns, so that they are easily available when needed. This also motivated us to create Peony and make it available on the Web.

3. The Peony Environment

Peony is a Web application that was conceived with the goal of supporting the search, application, and documentation of patterns used during the various stages of a software development process, as well as informing the test requirements needed to validate the proposed patterns. Peony is available at the following URL: <http://www.labes.icmc.usp.br:8180/alechan/>.

3.1. Main functionalities

Peony functionalities (around 168 different use cases) are divided into three groups: Feeding Peony, Managing Projects, and Managing Peony. The first group deals with activities for populating Peony database with available development processes (together with their stages, activities, and respective input and output expected artifacts), software patterns (together with their elements, solution structure, etc.), and associated test requirements. The present version of Peony supports only functional test criteria, namely the equivalence partitioning and the boundary value analysis, but other criteria could be included in posterior versions.

In the second group, activities regarding the project preparation and execution are conducted. Some phases or activities of a software process can be considered mandatory or optional. So, before instantiating a process (through a project) it is necessary to define its expected phases, activities, input, and output artifacts. This is done to adapt the process according to the project context. Then, the project is executed and, during it, Peony suggests candidate patterns to be applied and respective test requirements that can be used to validate them. For example, Fig. 1 illustrates a particular project being executed. Notice, in the left lower panel, the patterns suggested by Peony (more information about the pattern is supplied just by clicking on it).

The third group provides operations related to the maintenance of information contained in Peony, as for example users management and access control. All activities are performed using the Peony environment itself. Some elements are mandatory, while others are optional. There are also validation rules to maintain information consistency.

¹See for example the Pattern Catalog at <http://www.hillside.net>

Peony allows two types of users: software engineers and administrators, which can play several roles: project managers can include established development process, as well as their composing phases, activities, and input/output artifacts; pattern authors or users can include patterns and associate them to existing process phases and activities; system testers can include test requirements associated to patterns to help validating their instantiation; and developers (mainly project managers) can include new projects, assign them to a development process, customize the process according to the particular project context, and finally trace the project execution through Peony, which will suggest patterns according to the current phase/activity being conducted, as well as the associated test requirements. If developers decide to use a certain pattern in that phase/activity, they register this fact in Peony, which also allows the instantiation of the pattern solution to the particular project (specific Peony tags "Project Diagrams" and Project Software Patterns" are filled in).

Feeding Peony with patterns is our next challenge, as well as providing a mechanism to control the quality of patterns included. Peony was designed so that the organization can include patterns of a certain domain and make them available to all Peony users. Development processes can be included in a private repository, but can also be made available to other users. However, only the user who included the process (or the system administrator) can change it. When reusing a process, other users can customize the Mandatory/Optional attribute of phases/activities to allow skipping them during project execution.

3.2. Design and Implementation

Peony was developed based on the process for Web development proposed by Conallen [5] and the corresponding WAE (Web Application Extension) notation, which provides an extension to UML based on specific stereotypes for modeling Web application elements. This process is based on RUP (Rational Unified Process) [14] and on ICONIX Unified Process [20] and can be customized according to each particular project.

During its development, due to resources limitation (a master student played the roles of system analyst, programmer, tester, etc., with the advisor playing the role of customer), some stages had to be simplified, while others had to be included (for example, Conallen's process did not indicate an activity to develop the database models, but this was imperative in our case, due to the complexity of the conceptual model). Nevertheless, most artifacts were produced aiming at high quality documentation, and they are described in a master dissertation [4]. Examples of artifacts produced were: use case diagram, textual description of the most important use cases (around 60 in total), concep-



Figure 1. Peony - example of its graphical user interface.

tual model, entity-relationship model, activity diagrams for some use cases, sequence diagrams for the most important use cases, User eXperience (UX) models for most meaningful user interfaces. Figure 2 shows Peony concept model (simplified due to space restrictions).

Peony was implemented using Java [21] and the supporting frameworks Hibernate [19] for persistence and ZK [17] to allow rich user interface. The use of these frameworks facilitated the implementation of a quite complex Web application (the resulting code contains approximately 70 classes) by a very small team (only a student and her advisor) in a small period of time (around 8 months).

4. Conclusions and Future Work

The utilization of the Peony environment allows developers to better know, learn, and use software patterns in the creation of their applications, making it possible not only to use the patterns, but also having guidelines of how to validate them during VV&T activities. As future work we

intend to populate Peony with as many patterns as possible, especially Web application patterns, to help novice programmers on the implementation of Web applications. Peony is available on the Web, so any users who can test Peony and use it and give us feedback are welcome. The frequent usage of Peony will allow us to collect data about pattern utilization during the different stages of software development. Another future work will be the inclusion of other test criteria, feed Peony with other different development processes (at present it supports Connalen and the Unified Process), reinforce users to feed it with patterns and submit it to real usage in an industrial environment.

References

- [1] G. Bertollo, B. M. Segrini, and R. A. Falbo. Evolving the Definition of Software Processes in ODE. In *SBES'06 - Tools Session*, pages 109–114, Florianopolis/SC - Brasil, October 2006.

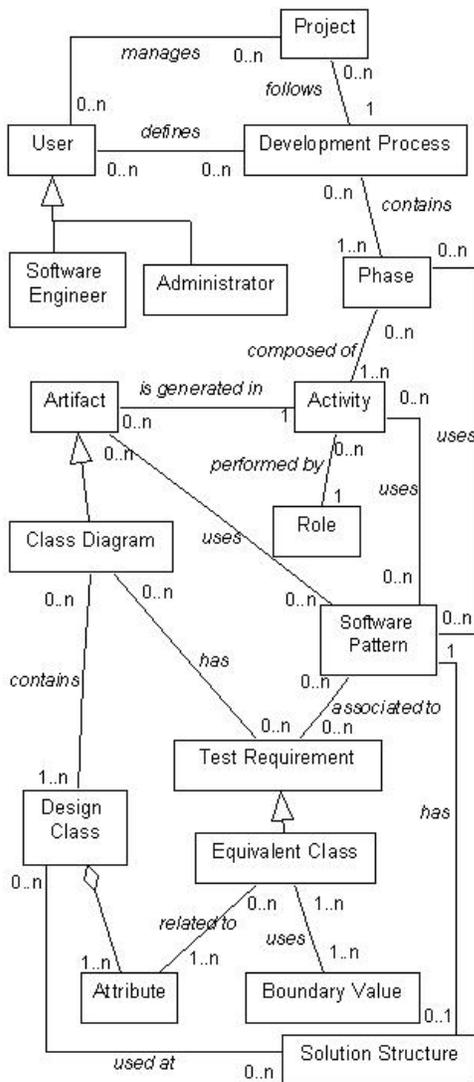


Figure 2. Peony Conceptual Model.

[2] D. Bonura, R. Culmone, and E. Merelli. Patterns for Web Applications. In *SEKE '02 - Proceedings of the 14th international conference on Software engineering and knowledge engineering*, pages 739–746, New York/NY - USA, 2002. ACM Press.

[3] M. I. Cagnin, R. T. V. Braga, F. Germano, A. Chan, and J. C. Maldonado. Extending Patterns with Testing Implementation. In *SugarLoafPlop'2005, V Conferencia Latino-Americana em Linguagens de Padres para Programao*, Campos do Jordo/SP - Brasil, Agosto 2005.

[4] A. Chan. Peony: a web environment to support development processes using software patterns and test requirements during applications' design (in Portuguese). Master's thesis, ICMC-University of Sao Paulo, Sao Carlos/SP - Brasil, February 2008.

[5] J. Conallen. *Buildind Web Applications with UML*. Addison-Wesley, 2nd. edition, 2002.

[6] J. Danculovic, G. Rossi, D. Schwabe, and L. Miaton. Patterns for Personalized Web Applications. In *EuroPloP'2001 - Proceedings of the 6th. European Conference on Pattern Languages of Programs*, Irsee - Germany, July 2001.

[7] R. A. Falbo, F. B. Ruy, J. Pezzin, and R. D. Moro. Ontologies and Semantic Software Development Environments (in Portugues). In *JIIISIC'04 - IV Jornadas Iberoamericanas de Ingeniera del Software e Ingeniera del Conocimiento*, Madri - Espanha, November 2004.

[8] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable of Object-Oriented Software*. Addison-Wesley, 2th. edition, 1995.

[9] F. Garzotto, P. Paolini, D. Bolchini, and S. Valenti. "Modeling-by-Patterns" of Web Applications. In *ER '99, Proceedings of the Workshops on Evolution and Change in Data Management, Reverse Engineering in Information Systems, and the World Wide Web and Conceptual Modeling*, pages 293–306, London, UK, November 1999. Springer-Verlag.

[10] F. Garzotto, P. Paolini, and D. Schwabe. HDM - a model-based approach to hypertext application design. *ACM Transactions on Information Systems*, 11(1):1–26, January 1993.

[11] M. Hakala, J. Hautamki, K. Koskimies, J. Paakki, A. Viljamaa, and J. Viljamaa. Architecture-Oriented Programming Using FRED. In *ICSE '01: Proc. 23rd International Conference on Software Engineering*, pages 823–824, Washington/DC - USA, May 2001. IEEE Computer Society.

[12] T. Isakowitz, E. A. Stohr, and P. Balasubramanian. RMM: A methodology for structured hypermedia design. *Communications of the ACM*, 18(8):34–44, August 1995.

[13] N. Koch and G. Rossi. Patterns for Adaptive Web Applications. In *EuroPloP'2002 - Proceedings of the 7th. European Conference on Pattern Languages of Programs*, Irsee - Germany, July 2002.

[14] P. Kruchten. *The Rational Unified Process: An Introduction*. Addison-Wesley, 2th. edition, 2000. 298 p.

[15] A. Lima, A. Costa, B. Frana, C. A. L. Reis, and R. Q. Reis. Flexible Management of Software Processes with the WebAPSEE Environment (in Portuguese). In *SBES'06 - SBES - Tools Sessions*, pages 97–102, Florianopolis/SC - Brasil, October 2006.

[16] F. Marinho, M. Santos, R. N. Pinto, and R. Andrade. Proposal of a Pattern Repository integrated to RUP (in Portuguese). In *SugarLoafPlop Proceeding 2003, The Third Latin American Conference on Pattern Languages of Programming*, pages 277–290, Porto de Galinhas/PE - Brasil, August 2003.

[17] Potix Corporation. ZK - Simply Ajax And Mobile. Online, 2007.

[18] R. S. Pressman. *Software Engineering: a Practioner's approach*. McGraw-Hill, 6th. edition, 2005.

[19] Red Hat Middleware, LLC. Hibernate Reference Documentation - Version: 3.2.2. Online, 2007.

[20] D. Rosenberg and K. Scott. *Use case driven object modeling with UML: a practical approach*. Addison-Wesley Longman Publishing Co., Inc., Boston/MA - USA, 1999.

[21] Sun Microsystems, Inc. JavaBeans TM. Online, August 1997.