Product Line Testing Group

This group has the task to investigate testing criteria, techniques and tools adequate to PL testing. At the very end, the aim is to contribute to the establishment of low-cost, efficient testing strategies in the context of PL development, ideally in the scope of a pre-defined PL development process. We would also like to conduct experimental studies with the perspective of evaluating the cost and benefits of developing products based on product lines against the traditional approaches. We consider in the scope of this study developing open educational and training modules to support the use and dissemination of the underlying concepts and tools. This research line is motivated by the fact that there are few initiatives of systematically addressing testing in the scope of product line production. It is also of our interest to explore other VV&T techniques, such as inspection and PL architecture evaluation. We will explore these ideas in the scope of two PL domains: meshing tools and interpreters, as discussed below.

First of all, we decided to provide to the Chilean partners the JABUTI testing tool family, developed at the ICMC-USP, and related training material, motivating the use of control and data-flow based testing criteria for OO and AO development. In a short period of time (2 months) the mutation based criteria will also be available in these tools.

In a Software product line (SPL) we can identify at least two important stages [6]: Domain Engineering (Core Asset Development) and Application Engineering (Product Development). The role of domain engineering is to produce common parts of the applications. The role of application engineering is to be a consumer of domain engineering producing the applications based on the common parts of the SPL [7].

Depending on the approach [3, 4], Domain Engineering has three or four stages, respectively. The additional one is Domain Testing. Similarly, one additional stage, Application Testing, is included in Application Engineering.

For Domain engineering the stages are:

1. Domain Analysis DA:
DA is the process through which the information used for developing all software systems within the SPL scope is identified, captured and organized with the purpose of making it available for reuse in future developments [5]. It has been identified as one of the most important factors for the success of software reuse [1].

2. Domain Design (DD):
DD is the process of developing a design model from the products of domain analysis and the knowledge gained from the study of software requirement/design reuse and generic architectures [6]. The DD encompasses all activities for defining the reference architecture of the product line. The reference architecture provides a common, high-level structure for all product line applications [4].

3. Domain Implementation (DI):

DI is the process of identifying reusable components based on the domain model and generic architecture. Using the domain knowledge gathered during domain analysis, and the generic architecture developed during the domain design, domain engineers acquire and, where necessary, create reusable assets which are catalogued into a component library for use by application engineers [6].

4. Domain Testing (DT):
DT is responsible for the validation and verification of reusable components. DT tests the components against their specification, i.e. requirements, architecture, and design artifacts. In addition, DT develops reusable test artifacts to reduce the effort for application testing [4].

We divided the main activities to achieve these goals in short, medium and long term.

In **short term**, we would like to conduct two systematic reviews in order to establish the ground for our initiatives. One would be to identify the testing activities and tools that have been explored for PL testing. The second one would be to identify the PL development processes and underlying VV&T activities. With these studies we hope we will be able to propose a VV&T strategy to a pre-defined or selected PL development process.

- Systematic Review of PL testing

Primary Question 1: Which techniques and criteria have been investigated or applied for PL testing?

Secondary Question 1: Among the techniques and criteria being investigated in the PL development context, which are the specific ones for PL?

Secondary Question 2: What are the defect types specific to PL development that have been identified, including taxonomy and defect models.

Secondary Question 3: What types of experimental studies have been conducted with relation to PL testing approaches?

Secondary Question 4: Which techniques and criteria have supporting tools? Are they open source?

Secondary Question 5: What are the open problems?

DEADLINE: September 2008

- Systematic Review of PL development processes

Primary Question 2: What are the PL software development processes and methods that have been investigated?

Secondary Question 1: Which are the VV&T techniques and criteria that have been proposed or used inside these processes?.

Secondary Question 2: What are the languages and models that have been used inside these processes?

DEADLINE: November 2008
(we should get together with the PL group (Thais, Sergio and Cecilia)

We also consider that it would be worthwhile to explore (**medium-term goals**) the adequacy of some previous work on testing in the scope of PL development and testing:

- regression testing:

Initiatives like coverage and modification based regression testing are worthwhile to be investigated to support the reuse of testing artifacts among the testing activities related to the products of a product line. Moreover, it seems interesting to investigate how traditional regression testing approaches would fit to the evolution and maintenance of a product line and related products. A “fast” bibliographic review shall be conduct.

DEADLINE: September 2008

- architecture based testing:

There are many initiatives to establishing testing based on architectures. We believe these works can be effectively explored for PL architecture testing and also for generating testing requirements for the related PL products. A “fast” bibliographic review shall be conduct.

DEADLINE: September 2008

- Define Product Line Testing Strategies (Test Bed Product Line):

The basic idea here is to think about a Product Line to generate Test Bed Products. This is a very new idea, as far as we now, and it should be further explored and clarified. We put it here just to register the idea.

- PL Architecture evaluation:

There is an on-going work at ICMC-USP, under Maldonado´s supervision looking at a metric suite to evaluate PL architectures. We would like to explore it in the two PL domain we will be investigating in this group.

- Open Didatic and Trainning Material development

Many researches have been conducted regarding to education and learning. In this context, one of the relevant activities is the development of educational modules. In a previous PhD thesis (BARBOSA 2006), we have discussed and investigated mechanisms to support the content modeling activity and the development process of such modules. Requirements and perspectives for conceptual, instructional and didactic modeling were identified. An integrated approach (IMA−CID) dealing with different perspectives related to the modeling content activity have been proposed. In the conceptual level, extended conceptual maps are applied. In the instructional level, the HMBS/Instructional model is established. In the didactic level, the HMBS/Didactic model is proposed. Regarding to the development process, systematic activities and tasks are established in the context of a standard process for the development of educational modules. Specialization and instantiation activities are also investigated. A maturity model – CMM/Educational – has been proposed. We intend to apply such results to develop educational and training modules in the scope of PL research and development.

Recently, we have investigated these approaches inside the QUALIPSO project. Qualipso project is one of the largest Open Source initiative funded by the European Commission, and is funded under EU´s sixth framework program (FP6), as part of the Information Society Technologies (IST) initiative. It is a unique alliance of European, Brazilian and Chinese ICT industry players, SMEs, governments and academics to help industries and governments fuel innovation and competitiveness with Open Source software. The aim is to leverage the Open Source Software development current practices to sound, well recognized and established industrial operations. A network of Open Source Competence Centers will make available the results of the Qualipso Project (www.qualipso.org). In Brazil, the Competence Center will be at the Universidade de São Paulo.

Our **long-term goals** would be investigating the VV&T activities in the two PL domain mentioned above: Messhing Tools and Interpreters.

- Establishing and evaluating testing strategies for Meshing Tool Domain PL

Meshes are used for numerical modeling, visualizing and/or simulating objects or phenomena [2]. A mesh is a discretization of a certain domain geometry. This discretization can be either composed by a unique type of element, such as triangles, tetrahedra or hexahedra, or a combination of different types of elements. Meshing tools generate and manage these discretizations.

Meshing tools are inherently sophisticated software due to the complexity of concepts involved, the large number of interacting elements they manage, and the application domains where they are used. Among others, these domains include engineering (e.g. mechanical and structural design) and medicine (e.g. surgery). Meshing tools complexity mainly relies on the components involved, as is the case for all scientific computing software, and not on the complicated distribution or concurrency because meshing tools are usually single process desktop applications.

- Looking at Intrepreters as a Product Line Domain:

Interpreters are a mean to specify programming language semantics. The benefit of using interpreters instead of compilers or virtual machines for this specification is their higher level of abstraction. Using interpreters is no longer necessary to deal with low level issues like machine code programming and program optimizations.

Interpreters make it easier to experiment with small variations of language semantics and consequently, easier to produce a new interpreter with slightly different semantics starting from a previously defined one. This is an interesting case due to the current trend towards domain-specific languages. The problem here is that testing all those tailored interpreters does not seem to be an easy task. In this workshop we conceived the idea of seeing this set of interpreters as products from the same product line. In this perspective, we intend to benefit from compiler testing experience to validate the interpreters. We also think that generic approaches for testing like mutation testing can be very useful in this case, for embedding facilities to test generation in the interpreters.

- Experimental Study definition

After defining processes and related VV&T strategies for the two above PL domains, we will carry out experimental studies, in bilateral collaboration, aiming at evaluating the cost and benefits of developing products based on product lines against the traditional approaches.

Requests for students interchanging

We think it would be effective to have bilateral technical visits. ICMC-USP intends to host Pedro and Rodolfo to improve the collaboration and the work in progress. In the other way around, ICMC-USP also intends to send students to Universidade do Chile. All the visits should be planned for the second semester of 2008, with a duration from 2 to 4 weeks.

[1] Guillermo Arango. A Brief Introduction to Domain Analysis. In Proceedings of the 1994 ACM Symposium on Applied Computing (SAC '94), pages 42-46. ACM Press, 1994.

[2] Rod W. Douglass, Graham F. Carey, David R. White, Glen A. Hansen,Yannis Kallinderis, and Nigel P. Weatherill. Current views on grid generation: summaries of a panel discussion. Numerical Heat Transfer, Part B: Fundamentals, 41(3-4):211-237, March 2002.

[3] Linda Northrop and Paul Clements. A Framework for Software Product Line Practice. Version 5.0, July 2007. http://www.sei.cmu.edu/productlines/framework.html.

[4] Klaus Pohl, Günter Böckle, and Frank van der Linden. Software Product Line Engineering. Foundations, Principles, and Techniques. Springer, August 2005.

[5] Rubén Prieto-Díaz. Domain Analysis: An Introduction. SIGSOFT Software Engineering Notes, 15(2):47-54, 1990.

[6] SEI. Domain Engineering, January 2007.
http://www.sei.cmu.edu/domainengineering/domain_eng.html.

[7] Antti Tevanlinna, Juha Taina, and Raine Kauppinen. Product Family
Testing: a Survey. SIGSOFT Software Engineering Notes, 29(2):12-17, March
2004.

Reference:
BARBOSA, Ellen Francine; A Contribution to the Modeling and to the Development Process of Educational Modules. PhD thesis, Instituto de Ciências Matemáticas e de Computação, University of São Paulo, São Carlos (SP), 2006, 270p. (In Portuguese).