ProLine-RM - Tool to Manager Crosscutting Framework Families

Rafael S. Durelli[1], **Valter V. de Camargo**[2]

[1]Computer Systems Department University of São Paulo
São Carlos, SP, Brazil.

[2]Computing Departament
Federal University of São Carlos (UFSCAR)
São Carlos, SP, Brazil.

rdurelli@icmc.usp.br[1], valter@dc.ufscar.br[2]

Abstract. *Abstract will be here.*

1. Introduction

2. Motivation

In the literature, it is possible find out a lot of researches related to CFF (COLOCAR REF). However, to the best of our knowledge, there is neither an approach nor a tool that shares, manages and provides full cycle of reuse of the CFF. Moreover, it is important a tool that supplies a graphical way to allow the engineer examines previously if the features available in one CFF fulfill the application requirements.

In order to overcome such limitations we have put forward a plug-in named Product Line-Repository Manager (ProLine-RM) which implements complete cycle of reuse of the CFF. Using this plug-in the CFF can be reused in a controlled way during the reuse to increase the quality of the developed applications. Furthermore, any manipulation in the CFF can be made through the intermediary of models, so we intend to raise the level of abstraction in which the engineer works.

3. ProLine-RM

In this section, we present the ProLine-RM tool, which is useful for providing full cycle of reuse of the CFF. The ProLine-RM has been developed upon the Eclipse Plugin License [1] (EPL). The use of the tool is twofold, the Domain Engineering (DE) phase where all artifacts are developed and upload to a repository, and the Application Engineering (AE) phase, where the reuse is done effectively. In order to illustrate the ProLine-RM in the DE phase and its functionalities we describe all activities necessaries to devise a "persistence" CFF. Similarly, to illustrate how the ProLine-RM assists the engineer in the AE phase we have developed an application that use the "persistence" CFF. These phases are described in the next two sections, respectively.

3.1. Domain Engineering

Figure 1 depicts a screenshot of ProLine-RM. In this figure the DE phase is exhibits by letter "A" to "C". In the first step the domain related to "persistence" has been studied.

[1] www.eclipse.org

The outcome of such study were the identification of both the common features and its variants of the domain.

After identifying the all features of the CFF the next activity is the development of the CFF effectively (see Figure 1(A)). For the purpose of accomplishing this activity we have used the approach described by Camargo and Masiero [2008], its aims is to assists and makes easier the develop of the CFF by using aspect oriented paradigm [Kiczales et al., 1997].

Afterwards, the feature model depicting all features related to the domain has to be modeled. Aiming to make easier the development of feature models, ProLine-RM provides a graphical way to assists the engineer devise them. Figure **??**(B) shows the feature model that we have developed. As can be seen, there are two mandatory features. The first one, called "Persistence" aims to introduce a set of persistence operations into application persistence classes (e.g., store, remove, update, perform queries). The second feature, named "Connection" is related to the database connection concern and identifies points in the application code where the connection must be opened and closed. This feature has variabilities, as for example Data Base Management System (e.g., MySQL, SyBase, Native and Interbase). There are two optional features as well. The former is called "Caching", which is responsible to deal with high-performance to get datas of the databases. The second, named "Pooling" is represented a set of database connections maintained by the databases.

After developing all artifacts, them have to be uploaded in a repository in order to be reused during the AE phase. Prior to uploading these artifacts, informations (e.g., Named of CFF, Author(s) and Description) associated with the "persistence" CFF has to be filled in. Figure 1(C) shows an example wherein the CFF is being uploaded.

3.2. Application Engineering

As stated previously, in the AE phase is wherein the reuse is started effectively.

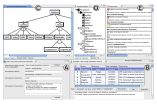

Figura 1. Screenshot of ProLine-RM

3.3. ProLine-RM Architecture

4. Related Work

5. Concluding Remarks

6. Acknowledgements

Referências

Camargo, V. V. and Masiero, P. C. (2008). A pattern to design crosscutting frameworks. In *Proceedings of the 2008 ACM symposium on Applied computing*, SAC '08, pages 759–764, New York, NY, USA. ACM.

Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C., marc Loingtier, J., and Irwin, J. (1997). Aspect-oriented programming. In *ECOOP*. Springer-Verlag.