Product-Line Instantiation Guided By Subdomain Characterization: A Case Study *

Patricia Pernich and Agustina Buccella and Alejandra Cechich and Maximiliano Arias and Matias Pol'la
GIISCO Research Group
Departamento de Ingeniería de Sistemas - Facultad de Informática
Universidad Nacional del Comahue
Neuquen, Argentina
patricia.pernich@gmail.com,
{agustina.buccella,alejandra.cechich}@fai.uncoma.edu.ar,
{ariasmaxi89,matpolla}@gmail.com
and
Maria del Socorro Doldan and Enrique Morsan
Instituto de Biología Marina y Pesquera "Almirante Storni"
Universidad Nacional del Comahue - Ministerio de Producción de Rio Negro
San Antonio Oeste, Argentina
{msdoldan,qmorsan}@gmail.com

Abstract

Reuse-oriented software methodologies have emerged to provide a set of rules or guides for development and evolution of software systems by reusing existing domain engineering artefacts such as requirements, architecture, components, test cases, etc. Domain engineering methods, such as software product lines engineering, aim at reducing development time, effort, cost, and complexity by taking advantage of the commonality within a portfolio of similar products. However, these methodologies are not based on specific domains, instead they cover a wide range of domains without considering particularities within them. Therefore, in this work we define a novel approach in which the creation of a software product line is guided by a subdomain analysis process, oriented to a particular case within the marine ecology domain. The methodology is then applied to a specific organization in order to create a product of the line.

1 Introduction

The Software Product Line Engineering (SPLE) [4, 12, 16] proposes a software development process by considering the main aspects and restrictions of the domains in which the lines are being built. The main characteristics involved in this discipline are [16]: *variability*, in which individual systems are considered as variations of a common part; *architecture-based*, in which the software must be developed by considering the similarities among individual systems; and a *two-life cycles approach*, in which two engineerings in every software product line process must be considered: *domain engineering* and *application engineering* [12, 16]. There exist several proposals in the literature describing different methodologies for developing software product lines [2, 4, 12, 16]. All of them propose a division into common and variable aspects of the product line, and a set of tasks or activities that must be done to specify and implement these aspects. For example, in [4, 12, 16] authors propose the same type of phases but with different names and activities. Thus, in every SPL project, identifying commonalities and variabilities of a particular problem domain is the key to improve functional and non-functional qualities and reduce product implementation time. However, the definition, representation and management of these common and variable requirements are still under research. For instance, there are several approaches proposing different ways to model variability in SPL based on two different paradigms: those based on feature oriented models [5, 9, 10] within the domain analysis area [1]; and those based on

* This work is partially supported by the UNComa project 04/E072 (Identificación, Evaluación y Uso de Composiciones Software) and by the PAE-PICT 2007-02312 (Métodos y Herramientas para Software Masivamente Distribuidos).

extensions of UML diagrams [7, 8, 12, 14, 15]. However, although these different approaches propose advantages with respect to different aspects of defining requirements, there is no standard way or set of specific rules to specify them [3, 13].

In addition, we have to take into account that the tasks of defining which requirements will be part of the line (with their variabilities) and which others will be product-specific are not easy and strongly depend on the domain involved. For example, in domains in which the level of specificity is higher, commonalities and variabilities should be, in theory, easier to define. This assumption can be analyzed in generic domains such as the geographic one. In this domain, the great number of implemented products share a set of common features[1] that are in general available to be used on all of them. For instance common services such as map panning and zoom, edition of geographic features, layer management, etc. can be easily extracted from any GIS. In addition, we can see another great number of features that are product-specific and are only implemented in some products. Also, the standard information defined by the Open Geospatial Consortium[2] (OGC) and the ISO Technical Committee 211[3] (ISO/TC 211, Geographic Information/Geomatics) provides a taxonomy of geographic services that can be used as a starting point to define these common services. However, although this standard is useful to understand the wide range of services every GIS is able to offer, these services are defined in a very generic and abstract way. Creating a software product line with these services would generate too many open issues that could be handled only by very complex variabilities. Therefore, it is necessary to delimit the set of geographic services to a specific subdomain, in our case, the *marine ecology* domain. The abstract services defined in the standard can be adapted to this subdomain in order to obtain and manipulate information useful for organizations working in this subdomain.

This work emerges as an extension of a previous work presented in [11] in which we have shown an implementation of an SPL in the marine ecology subdomain. In this work, we define a new methodology for creating a software product line for geographic information within the same subdomain. Geographic software here shares a set of common services that are essential for every application; therefore these common services must be identified and modeled as part of the product line together with different variations. To do so, we refine the services provided by the ISO 19119 std adapting them to the user' requirements in the domain. Also, we define a set of steps and rules used in the creation of an application framework to be used as a platform for each product of the line. The framework covers behavior that is common to all products and allows developers to add product-specific features. The methodology applies knowledge obtained in the subdomain in order to guide the activities for creating the SPL.

This paper is organized as follows: next section describes our methodology to create a subdomain-oriented software product-line within the geographic domain. Section 3 shows our application of the methodology to create a software product line for the marine ecology subdomain. Then, we describe the instantiation of the product line applied to a real project. Future work and conclusions are discussed afterwards.

2 A Subdomain-Oriented Software Product Line Methodology

Our development methodology combines advantages of several methodologies widely referenced in academy and industry [2, 6, 9, 12] and extends them in order to apply a subdomain view which guides the development of a software product line. Figure 1 shows the main activities of the methodology concerning the domain engineering phase. This phase is divided into two main analisys: *subdomain* and *organizational*. At the first analysis, we define three processes which impact directly on the activities defined at the second analysis. The gray rectangles in the figure denote the processes of the subdomain analysis and their influence into the activities of the organizational analysis. At the organizational analysis, the subdomain information is used to analyze organizations within the subdomain. The information modeled and implemented at the organizational analysis will be a subset of the information captured at the subdomain analysis.

Next, we briefly describe the main activities of the three processes defined in the subdomain analysis.

- *Information source analysis*: This process involves three sources to be considered within the subdomain: standards, existing applications and domain experts. Firstly, the standard information is obtained from the ISO 19119 standard in order to extract, classify and refine general services of GIS domain. Secondly, the existing applications correspond to an analysis of geographic tools that are currently used by the organizations. And finally, domain experts are

[1] In this work a feature describes the functional and quality characteristics of a system [2, 12]

[2] http://www.opengeospatial.org

[3] http://www.isotc211.org/

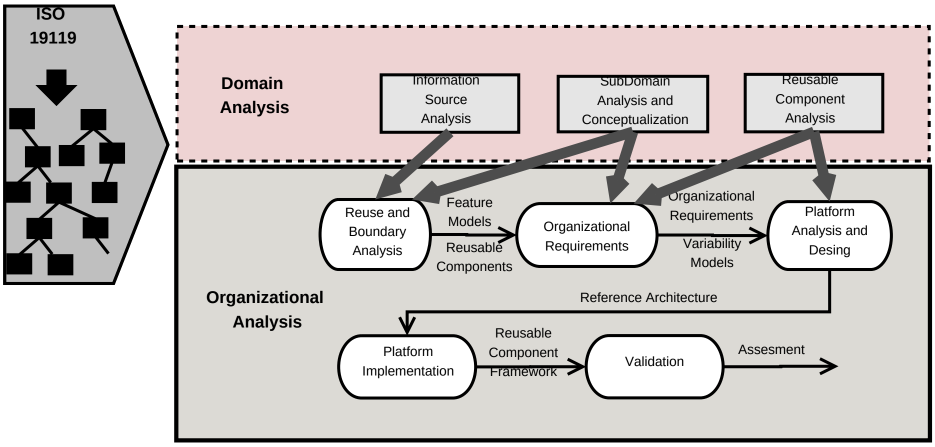

Figure 1. Activities of the Domain Engineering Process within a Geographic Subdomain

people experienced who can be currently (or not) working at these organizations and collaborating by providing useful information about different aspects to be considered. They provide specific information about their requirements and the way in which they perform their tasks.

- *Subdomain analysis and conceptualization*: Here, the information recovered in the previous process is used to analyze and organize the features[4] or services that the subdomain should offer. These features are defined by taking into account the standard services (ISO 19119 std) and the information provided by domain experts. In addition, in this process the subdomain must be conceptualized. Different software artifacts can be used here, such as class models, process models, etc.

- *Reusable component analysis*: This process identifies the set of reusable components that could be used to implement the features defined in the last process. Aspects as flexibility, evolution and maintenance must be carefully specified by means of the definition of the general structure of the subdomain. This process defines a reference model which will be refined by the activities at the organizational analysis. Thus, it must be created a preliminary structure composed of reusable components derived from the features obtained in the previous processes.

Following, we briefly describe the main activities at the organizational analysis (Figure 1) driven by the subdomain analysis processes (described previously).

- *Reuse and boundary analysis*: This activity defines the organizational boundary and commonality and variability features. Thus, by considering the features specified in the *subdomain analysis and conceptualization* process and the information from domain experts, the scope of the product line must be defined. Then, this activity analyzes which of the features can be implemented by geographic open source tools. In addition to functional features, other task in this activity is to identify non-functional requirements (quality aspects). In general, this type of requirements are modeled by the software engineers and will impact on the tasks of the next activities.

- *Organizational requirements*: In this activity, we use the information of the commonality and variability features identified in the last activity and the information provided by the *subdomain analysis and conceptualization* and *reusable component analysis* processes. The main goal here is to define the range of products and features that the line is able to implement. As our methodology follows a minimalist approach, only the features used in all products are part of the product line. This approach allowed us to fully implement only common features and let the product-specific features be implemented by each different organization. Thus, our software product line is then seen as a *platform* [16].

- *Platform analysis and design*: This activity builds the reference architecture based on the features defined in the previous activities and processes. The preliminary structure of reusable components defined in the *reusable component analysis* process must be reorganized and refined in order to perform two tasks. Firstly, the components are refined in

[4] In [2] a feature is defined as logical unit of behavior that is specified by a set of functional and quality requirements

order to add the variabilities together with the design decisions to implement them. Secondly, the features are reorganized into two sets of requirements to separate functional from non-functional (quality) needs. These sets will be the basis to define our architecture's components.

- *Platform implementation*: In this activity the components that are common for all products, that is, the components of the line, are implemented. This activity creates the application framework which will be used as a platform for each product of the line. The framework must also allow developers to add the product-specific features.

- *Validation*: There are several aspects to analyze within this activity. Firstly, some test cases must be defined in order to test the framework and the specification of the product line. Secondly, when a new product is developed, we must test this new instantiation (as the one we will describe in Subsection 3.1).

3 Building the Software Product Line at the Ecology Domain Level

In this section we describe the design and implementation of the subdomain-oriented software product line methodology within the *marine ecology subdomain*. In order to do so, we work in collaboration with two organizations working on this subdomain: the Instituto de Biología Marina y Pesquera "Almirante Storni"[5] (IBMPAS), and the Centro Nacional Patagónico[6] (CENPAT-CONICET). Both organizations are responsible for analyzing and storing information about sea surveys in three gulfs of the Argentinian Patagonia (San Matías, San Jorge and Nuevo Gulfs). Each survey, performed once a year (when it is possible), collects information about the population of specific species living in this area. This information is then used for spatial processing in order to obtain information about spatial distribution of data, population variation patterns in different scales, etc.

Next, in order to describe and analyze our experiences on the creation of the marine ecology SPL, we specify the main activities performed within the processes defined in Figure 1. By considering the *subdomain analysis*, in this case, the marine ecology subdomain, we performed the following activities:

[5]http://ibmpas.org/

[6]http://www.cenpat.edu.ar/

- *Eliciting requirements of domain experts and analyzing existing applications*: The experts working on the ecology marine organizations, provided us the first requirements that they needed to perform their daily activities. This was useful to define the first set of services within this subdomain. Next, we analyzed the existing applications by considering the geographic tools that are currently used in these organizations. In our analysis, we observed that few organizations had applications involving geographic information. They had used only office software tools in which almost all the tasks were made manually. In this way, we decided to analyze source software tools available on the Web that fulfill at least some of the requirements in the subdomain. In [11] we have classified and analyzed some of these tools. This analysis was useful to know which software components or applications can be reused to implement specific services.

- *Specializing the service taxonomy defined in the ISO 19119*: By considering the information provided by domain experts and the software tools available, we specialized the service taxonomy in order to define the set of services that are specific of the marine ecology domain. Table 1 shows part of these services in which *categories* and *service* columns are defined according to the standard. For brevity reasons we do not include here all of them.

- *Designing services by using different software artefacts*: In our work, we firstly defined the conceptual model which is used by all the services in order to implement their functionalities. In addition, for each service we defined a set of software artefacts (specifically, use cases and collaboration diagrams) to represent different aspects of the services.

- *Defining the preliminary reference model by designing reusable components*: By considering the particularities of the marine ecology domain and the n-tier architecture proposed in the ISO 19119 std., we defined a layered architectural style in order to improve modifiability and scalability aspects. It is composed of three main layers, *geographic model*, *geographic processing*, and *user interface*. In addition, for each layer we specified the components according to the services defined in Table 1. For instance, the second layer defines services involving processing; part of them are services S5-S7 described in table.

By considering the *organizational analysis*

Categories	Service	Specific Services
Geographic human interaction	S1. Geographic viewer	S1.1) a) Show zones. b) Show stations within a zone. c)... S1.2) a) Show/hide the zone layer...
	S2. Geographic feature editor	S2.1) a) Show a map with the location of zones. b) Show a map with the abundance of species...
Geographic model/information management services	S3. Feature access service	S3.1) a) Query zones of density of species. b) Query zones in which the population of species are higher than a specific value. c) ...
	S4. Catalog service	S4.1) a) Query and edit data in a catalog of geographic services. b) ...
Spatial processing services	S5. Proximity analysis service	S5.1) a) Obtain the location of stations within a specific zone. b)...
Temporal processing services	S6. Temporal proximity analysis service	S6.1) a) Obtain the number of specimens of specific species in a zone at different times. b) ...
Thematic processing services	S7. Change detection service	S7.1) a) Find changes among densities of species on different surveys. b) ...

Table 1. Part of geographic services required by the marine ecology subdomain and defined according to the ISO 19119

within the marine ecology subdomain, we performed the following activities:

- *Determining the costs and staff needed to build the SPL*: The specific services defined in the last activities and the actual situation of the organizations in this subdomain, determined that the costs and staff needed were analyzed by taking into account two main phases. In the first one, we analyzed the aspects needed to implement the product line and its supporting application framework. In the second phase, we analyzed what we needed to implement future product lines by instantiating the framework. The output of this activity was an organizational model considering all these aspects.

- *Defining the set of geographic open source tools to be used*: Based on the classification and analysis of the geographic tools performed in the subdomain analysis and the set of services to be implemented we selected the definitive set of these tools to implement them. All this information was put in a feature/tool matrix in order to visualize which open source tools implement which features and define the future component/application reuse.

- *Defining the scope of the line and designing variability*: The task here is to create a product/feature matrix indicating the services required by each product. Then, by using this matrix, the services to be part of the line must be derived. For example, in our SPL, services S1-S3 and S5-S6 are part of the product-line and S7 is a product-specific feature that will be implemented only by Product 1 (see Table 1). Then, service S10[7] is only implemented by Product 2. In addition, within each feature we determined the commonality and variability models. For example, for the feature S2.1, the variability model described the two variants of editing and seeing geographic features - by using maps or tables.

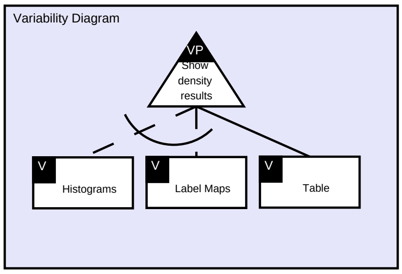

Figure 2. Variability model item associated with the Sequence Diagram of the service S7.1: *Find changes among densities of species*

- *Adding and designing variability*: We refined the conceptual model and the models used to represent each service by adding the variabilities included on each of them by means of variability models[8]. Figure 2 shows the variability model associated with the S7.1 service. In the figure we observe different representations of the returned data about the distributions of species in different surveys. In the variability model, the data are always shown in a tabular way and histograms and labels are alternative choices.

- *Designing the reference architecture of the line*: The reference model defined in the

[7]S10 analyzes the advance of the undaria pinnatífida seaweed and its relation with the population of echinoderms

[8]We used the notation of variability models proposed in [12], called *orthogonal variability model*, due to their clarity on defining variability over UML software artifacts.

subdomain analysis was reorganized and refined in order to design the reference architecture. Figure 3 shows this model together with which services (defined in Table 1) are implemented by which components and the dependences among them. For instance, the change detection component (of the geographic processing layer) implements the service S7.1 and the geographic statistics component implements services S13.1 and S13.2[9]. The components colored in gray are components that are part of the platform and the others are specific for one o more products in the line.

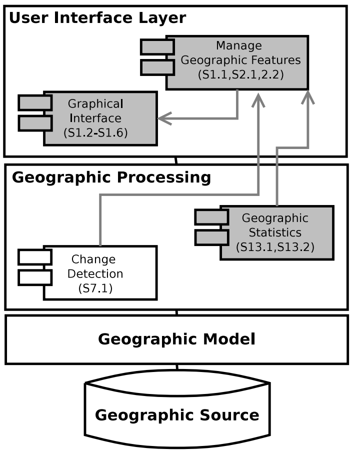

Figure 3. Reference architecture of the SPL for the marine ecology domain

- *Implementing the platform*: In a first prototype [11] we had built a software platform with a set of 20 common services approximately by using PostGIS[10], GeoServer[11] and OpenLayers[12] on each level of the reference architecture. Then, we instantiated this platform in order to create a new product with three product-specific services. This instantiation is described in the next subsection.

3.1 Instantiating the Product Line: A Case Study

In this section we show a instantiation of the SPL created specifically for the marine ecology subdomain. We build the Product 1 containing features of the product line plus a set of product-specific ones (S7). The other products in the software line will belong to other organizations within the marine ecological domain (Instituto Argentino de Oceanografía[13], Centro Nacional Patagónico[14], and Laboratorio de Moluscos y Crustáceos belonging to the University of Mar del Plata).

Product 1 emerged from a project between the GIISCO research group[15] and the Instituto de Biología Marina y Pesquera "Almirante Storni"[16] (IBMPAS). IBMPAS is responsible for analyzing and storing information about sea surveys in the San Matías Gulf, Patagonia Argentina. Each survey, performed once a year (when it is possible), collects information about the population of specific species living in this area. This information is then used for spatial processing in order to obtain information about spatial distribution of data, population variation patterns in different scales, etc.

In order to instantiate and develop the Product 1 we have performed the following steps:

- We have defined the services of Table 1 according to the requirements of the product. The features were refined to include the specific layers required by the IBMPAS. Thus, in this product we implemented several layers including surveys, sea zones[17], and stations[18].

- We have instantiated the variability models associated to the features (Table 2). For instance, the feature S7.1 in this product is implemented by using labels (not by using histograms).

- Finally, we have created the architecture based on the reference architecture defined in the domain engineering process (Figure 3). This architecture contains the components defined for the SPL and those components specifically created for this product (S7).

Part of Product 1 is already implemented. As an example, we describe here two of the services implemented – *change detection* and *query geographic feature* services. The first one is shown in Figure 4 and returns information about the places (stations) in which a species (Viera Tehuelche) is found in different surveys (year 87 in yellow and year 96 in violet) in different zones (polygons in green). This service is very useful to analyze migration movements of species. Feature S7.1 is implemented in this service. The other service, query geographic feature, can be also seen in Figure 4.

[9] S13 generates statistics from geographic features providing several ways to represent the information (charts, tables, etc.)
[10] http://postgis.refractions.net/
[11] http://geoserver.org/
[12] http://openlayers.org/
[13] http://iado.criba.edu.ar/web/
[14] http://www.cenpat.edu.ar/
[15] http://giisco.uncoma.edu.ar/
[16] http://ibmpas.org/
[17] A zone is a maritime area bounded and defined with a specific name in the gulf.
[18] A station is a geographic point located within one of the defined zones. In this location the measures of population of species are obtained.

The service presents a description table in the bottom when one or more zones are selected (in this case two zones were selected). This service is the implementation of feature S2.1 with the *table* variability instantiated.

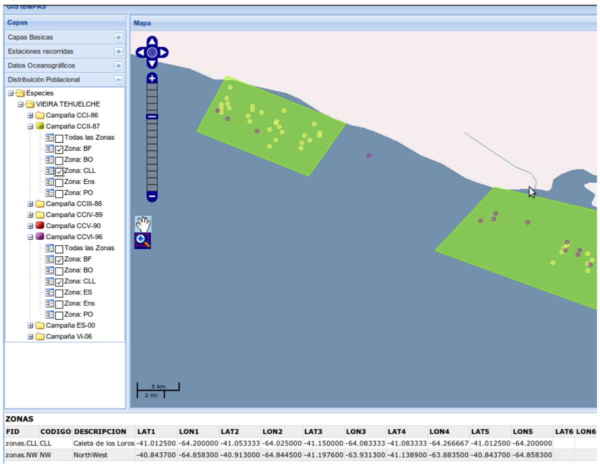

Figure 4. A map showing the *change detection* **and** *query geographic feature* **services**

Product 1 is partially implemented and available on Internet at: `http://geoserver.ods.org/geoserver/www/webgissao/index.html`.

4 Conclusion and Future Work

In this work we have defined a methodology oriented to create a software product line in the marine ecology domain. Our work emerges as a solution to different organizations within this domain. Although a great effort must be put initially for creating the product line, benefits appear when creating each new product. In this paper we have shown a real case study in which we have instantiated the line in order to create a specific product for the IBMPAS. The benefits of this model of development have not been directly measured, but one indicator of the success is that the development cost was drastically reduced.

As future work, the methodology and the framework need more validation, but we are aware that developing management guidelines is also crucial for successfully applying the approach. Within this line, we are developing a supporting tool (as Eclipse plug-in) to interact with software engineers and developers in the process of creation of a new product in the line.

References

[1] G. Arango. Domain Analysis. In J. Marciniak, editor, *Encyclopedia of Software Engineering*, volume 1, pages 424–434. WILEY, 1994.

[2] J. Bosch. *Design and use of software architectures: adopting and evolving a product-line approach*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 2000.

[3] L. Chen and M. A. Babar. A systematic review of evaluation of variability management approaches in software product lines. *Information and Software Technology*, 53(4):344 – 362, 2011.

[4] P. Clements and L. Northrop. *Software Product Lines : Practices and Patterns*. Addison-Wesley Professional, August 2001.

[5] K. Czarnecki and U. Eisenecker. *Generative programming: methods, tools, and applications*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 2000.

[6] K. Czarnecki, S. Helsen, and U. W. Eisenecker. Formalizing cardinality-based feature models and their specialization. *Software Process: Improvement and Practice*, 10(1):7–29, 2005.

[7] L. Dobrica and E. Niemelä. A uml-based variability specification for product line architecture views. In *Proceedings of the Third International Conference on Software and Data Technologies, ICSOFT*, pages 234–239, Porto, Portugal, 2008.

[8] H. Gomaa. *Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 2004.

[9] K. Kang, S. Cohen, J. Hess, W. Nowak, and S. Peterson. Feature-Oriented Domain Analysis (FODA) Feasibility Study. Technical Report CMU/SEI-90-TR-21, Software Engineering Institute, Carnegie Mellon University Pittsburgh, PA., 1990.

[10] K. Kang, S. Kim, J. Lee, K. Kim, G. Kim, and E. Shin. FORM: A feature-oriented reuse method with domain-specific reference architectures. *Annals of Software Engineering*, 5:143–168, 1998.

[11] P. Pernich, A. Buccella, A. Cechich, S. Doldan, and E. Morsan. Reusing geographic e-services: A case study in the marine ecological domain. In *Software Services for e-World*, volume 341 of *IFIP*, pages 193–204. Springer Boston, 2010.

[12] K. Pohl, G. Böckle, and F. Van der Linden. *Software Product Line Engineering : Foundations, Principles and Techniques*. Springer, September 2005.

[13] I. Reinhartz-Berger and A. Tsoury. Specification and utilization of core assets: feature-oriented vs. uml-based methods. In *Proceedings of the 30th international conference on Advances in conceptual modeling*, ER'11, pages 302–311, Berlin, Heidelberg, 2011. Springer-Verlag.

[14] B. Rumpe and F. Robert. Variability in uml language and semantics. *Software and Systems Modeling*, 10(4):439–440, 2011.

[15] Z. T. and J. Jézéquel. Software product line engineering with the uml: Deriving products. In *Software Product Lines*, pages 557–588. 2006.

[16] F. Van Der Linden, K. Schmid, and E. Rommes. *Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

Received: June 2012. Accepted: August 2012.