

Co-evolution Analysis for Software Product Lines

Anissa Benlarabi, Amal Khtira and Bouchra El Asri

IMS Team, Laboratory SIME, ENSIAS, University Mohamed V, Rabat, Morocco

Keywords: Software Product Lines, Co-evolution, Cladistics Classification.

Abstract: The purpose of our approach is to study the co-evolution of the platform and the products of software product lines. Because the platform must be able to derive all the family products, products are not allowed to evolve independently from the platform, thus the propagation of the products changes must be managed efficiently. Instead of focusing on the change impact analysis we propose an approach to compare the evolution histories of the products and the platform illustrated through evolutionary trees built using the biological technique cladistics. This comparison yields important results concerning the change propagation. In this paper, we introduce the use of cladistics for software product lines to build evolutionary trees for platform and products, then we elaborate a mathematical analysis to compare these trees, afterwards we validate this work through a case study (mobile media software product lines). We also provide the design of an automated tool.

1 INTRODUCTION

Today the software development tends to reuse the knowledge more than building software from scratch. The software product line (SPL) engineering is the most widespread software reuse approach, it consists in building a common platform for a set of products dedicated for a business domain (Pohl et al., 2005). The main advantage of this approach is the improvement of the productivity by bringing to customers products adapted to their needs at a reasonable price and a short time (Clements et al., 2002). However, the customers requirements evolve continuously, thus the (SPL) must be adapted to cope with customers needs.

Among the most frequent SPL evolution problems, we mention the change propagation problem. this issue occurs when the products evolve independently from their common platform. In fact, if many changes are applied to the products without propagation to the common platform, the SPL will not be able to derive the family product. Hence, the SPL will fall into the aging phenomenon. This issue has been tackled by many approaches, specifically the traceability approaches (Anquetil et al., 2010) (Goknil et al., 2011) that use the links between the SPL artifacts to identify the impact of the change, the evolution modeling approaches (Romero et al., 2013)(Ajila and Kaba, 2008) that separates the evolution phase to elementary operations for better understanding of the impact. These approaches focus mainly on the change

understanding and the impact analysis and rely essentially on human knowledge (Benlarabi et al., 2014b).

Instead of focusing on the change impact analysis activity we proposed a new approach for co-evolution analysis in SPL, which consists in modelling the evolution history of the platform and the derived products in evolutionary trees and compare them to understand how they impacted each other during their evolution and also to discover the changes of products that were not propagated to the platform with the aim of propagating them to the platform. The comparison of evolutionary trees is a prominent technique used in biology to study the co-evolution of organisms (Ehrlich and Raven, 1964), it allows for understanding the history of life. To model the evolution of the SPI we use the cladistics classification (CC) (Brinkman et al., 2004). We illustrated our approach through a case study on the mobile media SPL (Tizzei et al., 2011).

In this paper the focus is on the summary of our co-evolution modelling approach and the elaboration of an automated tool. The remainder of this paper is structured as follows. Section 2 introduces the software evolution problem. Section 3 reminds the biological co-evolution principal, the cladistics method and the analogy between SPL co-evolution and biological co-evolution. Section 4 presents our approach to study the co-evolution in SPL. Section 5 illustrates the approach through the mobile media case study. Section 6 investigates the design of an automated tool and the section 7 concludes the paper.

2 SPLs EVOLUTIONS

The SPL engineering consists in building a platform composed of common assets for a set of products. By reusing these assets (Clements et al., 2002), customers get products adapted to their needs at reasonable price and short time. However, the users requirements evolve over time, hence the SPL must be adapted continuously to cope with the user requirements changes. In this section we present the SPL evolution propagation problem. Thereby, we discuss the previous research and we introduce our approach.

2.1 SPLs Evolution

Software evolution is an important activity because it must accompany the evolution of business requirements. Compared to single software evolution, the SPL evolution is more difficult since the change must be propagated in the two levels: the level of the products and the level of the platform. Hence, if a change concerns an application product, it could not be all the time propagated to the platform. Thereby, if products are changed frequently and independently, the domain artifacts of the SPL risk to be out of date and instead of having a set of product derived from the same domain artifacts, we will have a set of independent software.

2.2 Change Propagation Problem

The great majority of works done to deal with the change propagation problem such as the traceability (Anquetil et al., 2010)(Goknil et al., 2011) or the evolution modelling (Romero et al., 2013)(Ajila and Kaba, 2008) approaches focus on the change impact analysis activity because they consider that this problem is mainly caused by a shortage in this activity. Regarding the traceability approaches, they rely on links defined by the developers to determine the impact of the change, while the evolution modelling approaches separates the change to elementary operations and identify the impact of each operation based on a database of relationships between different components. Both approaches presents shortcomings. Firstly they rely on human knowledge and use data that have to be manually provided which is error prone and too expensive in practice. In addition, they perform only pre-analysis of the change.

We aim to overcome the presented shortcomings by studying the co-evolution of the products and the platform. Through our approach (Benlarabi et al., 2014b) we improve the change understanding by summarizing the evolution history in an evolutionary

tree, we also define the impact of the change by comparing the evolutionary tree of the platform with the evolutionary tree of each products, and we help correcting the divergences between the platform and the products.

3 CO-EVOLUTION FOR SPLs

In this section we introduce the co-evolution of organisms with concrete examples from biology, then we present the technique used to study the co-evolution in biology and we discuss the analogy with SPL co-evolution.

3.1 Co-evolution in Biology

According to Erlich (Ehrlich and Raven, 1964) the co-evolution is the examination of patterns of interaction between two major groups of organisms with a close and evident ecological relationship. The co-evolution of host-parasite is a famous example of biological co-evolution. Because parasites cause damages to their hosts, hosts develop new capacities to resist to their parasites however parasites also develop capacities to overcome this resistance (Anderson and M.May, 1982). Hence, a clearer understanding of hostparasite co-evolution will point to new possibilities for organic farming and reduce the application of ecologically harmful chemicals.

3.2 Co-evolution Analysis Through Cladistics

Co-evolution analysis consists practically on comparing the evolution histories of populations depicted by their phylogenetic trees (Fitch and Margoliash, 1967). the phylogenetic tree of a population is a branching diagram showing the inferred evolutionary relationships among their members on the basis of the similarities and differences in their physical or genetic characteristics. There are two main methods to create a phylogenetic tree (Fitch and Margoliash, 1967): phenetic which is DNA sequence based method, and cladistics which is character-based method. We choose cladistics method (Brinkman et al., 2004) because we rely on features. It consists in the following steps:

Organisms selection: the organisms of the classification should be necessarily related by ancestor-descendant relationships.

Characters extraction: the characters include physical and behavioral characters that allow to make distinction between organisms.

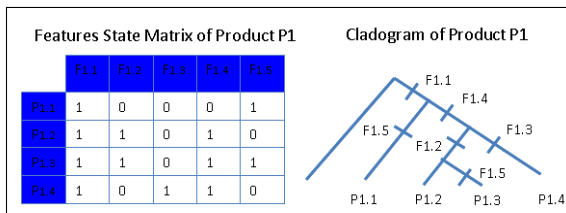


Figure 1: SPL and Product Cladograms.

Characters matrix construction: this matrix is filled by numbers that represents the codification of the states of the characters in organisms.

Cladograms drawing: the cladogram is a tree. All the organisms are in the endpoints of the tree, each node regroupes shared characters between all the next organisms.

3.3 Cladistics Classification for SPLs

The proposed approach aims at examining the co-evolution of domain and application feature models in SPLs, similarly to biology we investigate the co-evolution of two major groups of organisms that are closely related, the first group is composed by the common platform and the second group is composed by the derived products. All the populations members are derived from the same ancestor which is the first release of the common SPL platform. Thus, we study the interaction between the two populations on the basis of the history of their evolution. The input of our study is the feature models of the different releases of the platform and the products, otherwise, for a product we gather all the feature models of its releases then we build the evolutionary tree of the product on the basis of the models by applying the CC method. We repeat this operation for each product and also for the common platform in order to produce their evolutionary trees. Thereby we proceed to the comparison of the resulted trees.

4 CO-EVOLUTION OF DOMAIN AND APPLICATION ENGINEERING

Similarly to biology the characters of the products are the features and the variants of the SPL. In our previous work (Benlarabi et al., 2014b) we introduced a new approach for establishing the evolutionary trees of the platform and the derived products in based on cladistics classification, we also proposed a mathematical analysis to compare these trees and to find out the products changes that were not propagated to the

platform. In this section, we give an overview of our proposal.

4.1 Cladistics Application for SPLs

The first step of the CC method is the populations identification. In our study we identified many populations, each one is a set of software systems derived from the same ancestor software. The first population is composed by the releases of the common platform, while the other populations are composed by the different releases of the products. The second step is characters extraction, the characters are the features. We extract the independent features of each population. For example, Considering the following population of the product P1 composed by n versions $\{P1.1, P1.2, P1.3, \dots, P1.n\}$, the set of its characters is $\{F1.1, F1.2, F1.3, \dots, F1.m\}$ where m is the number of the independent features of the population.

After the extraction of characters we construct the characters states matrix of each population, each feature has two states, the primitive state 0 which denotes the non-existence of the feature in the release, and the derived state 1 which denotes its existence. in Fig. 1 we represent an example of the features states matrix of the population of the product P1. The last step is the evolutionary tree construction, in this step we draw a branching tree of the population based on the characters states matrix, in the Fig. 1 figures the resulted cladogram of the matrix of our population example

4.2 Perfect Co-evolution Hypothesis

By investigating the similarities and the divergences between the cladogram of the platform and the cladogram of each product, we can deduce the changes happened to the products that were not propagated to the platform. The branches of a cladogram represent the set of features of the population. The imperfect evolution is caused by branches that exist in one cladogram and are absent from the other. However, we eliminate the imperfection caused by branches that exist in the SPL cladogram and are absent from the products cladograms because features of the SPL could not be all used in the products. We express mathematically the relationship between the cladogram of a product and the cladogram of the platform on the basis of their features states matrices. In this subsection we formalize our hypothesis, then in the following subsection we proceed to the comparison and we discuss the restoration of missed features to the platform.

We consider two populations P1 and P2. P1 consists of k releases of the SPL and P2 consists of k releases of a derived product. By extracting features of the two populations we will obtain the sets $A1$ composed of n features and the set $A2$ composed of m features respectively for P1 and P2:

$$\begin{aligned} A1 &= \{F1i \text{ while } i \in \mathbb{N}, i \leq n\} \\ A2 &= \{F2i \text{ while } i \in \mathbb{N}, i \leq m\} \end{aligned}$$

We stated the following hypothesis in our previous work as follow:

(H0) Features Independence. In the set of features $A1$ and $A2$, the features are independent.

$$\begin{aligned} \forall i, j \in \mathbb{N}, i \leq n, j \leq n, F1i \neq F1j \\ \forall i, j \in \mathbb{N}, i \leq m, j \leq m, F2i \neq F2j \end{aligned}$$

(H1) Domain Features Sufficiency. Each feature of the set $A2$ has a corresponding feature in the set $A1$.

$$\forall j \in \mathbb{N}, j \leq m, \exists i \in \mathbb{N}, i \leq n / F2j = F1i$$

(H2) Features Exclusion. This hypothesis is deduced from the combination of **H0** and **H1**. Each feature of $A2$ has only one corresponding feature in $A1$.

$$\left\{ \begin{array}{l} \{F2i, F2j/i, j \leq m\} \subset \{F1x, F1x/x, y \leq n\} \\ F2i = F1x \end{array} \right\} \Rightarrow F2j = F1y$$

4.3 Perfect Co-evolution Restoration

On the basis of the hypotheses **H0** and **H2**, we deduce that the relationship between the two cladograms of P1 and P2 must respect the following inequality:

$$B \times A1 \geq C \times A2$$

Where B is the matrix ($k \times n$) of features states of P1, k is the number of taxa of P1 and P2, C is the matrix ($k \times m$) of features states of P2, $A1$ is the features vector ($n \times 1$) of P1 and $A2$ is the features vector ($m \times 1$) of P2.

After developing the inequality, it is reduced to a set of k inequalities:

$$\left\{ \begin{array}{l} \sum_{i=1}^n b_{1,i} \times F1_i \geq \sum_{j=1}^m c_{1,j} \times F2_j \\ \sum_{i=1}^n b_{2,i} \times F1_i \geq \sum_{j=1}^m c_{2,j} \times F2_j \\ \vdots \\ \sum_{i=1}^n b_{k,i} \times F1_i \geq \sum_{j=1}^m c_{k,j} \times F2_j \end{array} \right.$$

By considering the hypothesis **H1**, the inequalities can be transformed to a set of k equalities because the difference between the two sides of each inequality i where $0 < i \leq k$ will be equal to the set of features of the SPL version P1i that did not exist in the product version P2i. The number s represents the difference between n the number of features of P1 and m the number of features of P1 ($s = n - m$):

$$B \times A1 - C \times A2 = \begin{pmatrix} d_{1,1} & \cdots & d_{1,s} \\ d_{2,1} & \cdots & d_{2,s} \\ \vdots & \ddots & \vdots \\ d_{k,1} & \cdots & d_{k,s} \end{pmatrix} \times \begin{pmatrix} F3_1 \\ \vdots \\ F3_i \\ \vdots \\ F3_s \end{pmatrix}$$

The vector of features ($F3_1, F3_2, \dots, F3_s$) represents the set $A3$ which is given by subtracting $A2$ from $A1$ ($A3 = A1 - A2$), and the entries $d_{i,j}/i, j \in \mathbb{N}, 0 < i \leq k, 0 < j \leq s$ of the matrix D must be equal to 0 if the feature $F3_j$ exist in the SPL version P1i or 1 if the opposite is true.

In order to repropagate the identified changes to the SPL we propose the algorithm of restoration (Benlarabi et al., 2014b) in which we add to $A1$ all the feature of $A3$ that are absent from $A1$, these features represent the product features that are not derived from the domain features of the SPL. Then we extend the features states matrix of the SPL P1 with these features.

5 CASE STUDY

We performed a case study on the mobile media SPL (Benlarabi et al., 2014a). It manipulates photo, music, and video on mobile devices such as mobile phones and has many derived products (Tizzei et al., 2011). Firstly, we constructed a population of the SPL formed by its different releases, it is described in the table 1. We construct also a population for the derived product. We started by applying our approach on the Mobile Media and one of its derived product.

5.1 Cladistics Classification for Mobile Media SPL

We constructed two populations, the first one P1 is formed by seven versions of the SPL mobile media, while the second one P2 is formed by seven versions of the derived product. In Tables 1 and 2, respectively, we give a detailed description of the populations P1 and P2, which we constructed on the basis of the feature models of the different releases of P1 and P2: Each population is composed of six organisms, P1 has 19 features and P2 has 16 features:

$$\begin{aligned} A1 &= \{F1_i \text{ while } i \in \mathbb{N}, i \leq 19\} \\ A2 &= \{F2_i \text{ while } i \in \mathbb{N}, i \leq 16\} \end{aligned}$$

The features states matrices of the two population are constructed on the basis of Table 1 and Table 2. We present in Table 3 an example of the matrix B of the mobile media SPL. The matrix C of the products is constructed similarly. We constructed cladograms on

Table 1: Mobile Media Platform Population.

V1.0	The first release of the mobile media SPL, this release encompasses the following features: Manage photos ($F_{1,1}$), Create album ($F_{1,2}$), Delete album($F_{1,3}$), Create media ($F_{1,4}$), Delete media ($F_{1,5}$), View media ($F_{1,6}$), Sort media ($F_{1,7}$), Edit media label ($F_{1,8}$)
V1.1	Release 2: the features Set favorites ($F_{1,9}$) and See favorites ($F_{1,10}$) were added
V1.2	Release 3: the feature Copy media ($F_{1,11}$) was added
V1.3	Release 4: the features Send media ($F_{1,12}$) and Receive media ($F_{1,13}$) were added
V1.4	Release 5: the feature Add music media management ($F_{1,14}$) was added
V1.5	Release 6: the features Add video media management ($F_{1,15}$), Capture videos ($F_{1,16}$) and Capture photos ($F_{1,17}$) were added
V1.6	Release 7: the features Play videos ($F_{1,18}$) and Play music ($F_{1,19}$) were added

Table 2: Derived Product Population.

V2.0	The first release of the product, this release encompasses the following features: Manage photos ($F_{2,1}$), Create album ($F_{2,2}$), Delete album($F_{2,3}$), Create Photo ($F_{2,4}$), Delete Photo ($F_{2,5}$), View Photo ($F_{2,6}$), Sort media ($F_{2,7}$), Edit media label ($F_{2,8}$)
V2.1	Release 2: the features Set favorites ($F_{2,9}$) and See favorites ($F_{2,10}$) were added
V2.2	Release 3: the feature Copy media ($F_{2,11}$) was added
V2.3	Release 4: the features Send media ($F_{2,12}$) and Receive media ($F_{2,13}$) were added
V2.4	Release 5: the feature Print photo ($F_{2,14}$) was added
V2.5	Release 6: the feature Capture photos ($F_{2,15}$) was added
V2.6	Release 7: the feature Share photo in social websites ($F_{2,16}$) was added

Table 3: Features States Matrix B of the Mobile Media SPL.

B	$F_{1,1}$	$F_{1,9}$	$F_{1,11}$	$F_{1,12}$	$F_{1,14}$	$F_{1,15}$	$F_{1,18}$
	$F_{1,8}$	$F_{1,10}$		$F_{1,13}$		$F_{1,17}$	$F_{1,19}$
V0	1	0	0	0	0	0	0
V1	1	1	0	0	0	0	0
V2	1	1	1	0	0	0	0
V3	1	1	1	1	0	0	0
V4	1	1	1	1	1	0	0
V5	1	1	1	1	1	1	0
V6	1	1	1	1	1	1	1

the basis of these matrices by grouping releases together based on their shared characters. In this step we used the biological tool PHYLIP to generate the coordinates of the evolutionary trees of P1 and P2 from their features states matrices. We used the output files generated by PHYLIP in the online tool Phyfi to generate cladograms that correspond to the coordinates of these file. After generating the two cladograms we proceed to the co-evolution analysis of the two populations

5.2 Co-evolution Analysis of Mobile Media SPL

After applying the mathematical analysis we described before, we noticed that two imperfections were detected after the calculation. They are caused by the two features $F_{2,14}$ "Print photo" and $F_{2,16}$ "Share photo in social websites". The two features exist in the product and are absent from the SPL. Finally we obtained the following formulate.

$$B \times A1 - C \times A2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & -1 & -1 & 0 \end{pmatrix} \times \begin{pmatrix} F_{1,14} \\ F_{1,15} \\ F_{1,16} \\ F_{1,18} \\ F_{1,19} \\ F_{2,14} \\ F_{2,16} \end{pmatrix}$$

The vector A3 is composed by the following features: $F_{1,14}$ "Add music media management", $F_{1,15}$ "Add video media management", $F_{1,16}$ "Capture videos", $F_{1,18}$ "Play videos", $F_{1,19}$ "Play music", $F_{2,14}$ "Print photo", $F_{2,16}$ "share photo in social websites".

6 AUTOMATED TOOL

To implement our approach we have chosen to design a tool composed of three main functional blocks, the first block generates the features states matrix of a population, the second block compare two cladograms using their features states matrices and vectors of features and generates a log file containing the result of the comparison. and the third block proposes on the basis of the generated log file the features that may be added to the platform to correct the imperfect co-evolution.

6.1 Features States Matrix Generator

In this step we use the FeatureIDE tool (Kastner et al., 2009) to model our case study feature models. It is an open source framework for software product line engineering, the main advantage of this tool is the possibility to generate the XML scheme of the feature model. in Fig. 2 we illustrate a domain feature model of the mobile media software product line.

Once the xml files are generated, this block integrate the features reference file and also the generated files, then it looks for the features in each file and generate the matrix in a flat file in the accepted input format for the PHYLIP tool.

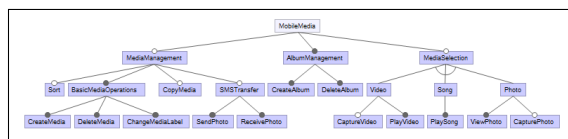


Figure 2: Mobile Media SPL feature model.

6.2 Co-evolution Comparator

The second block consists in comparing two input matrices, by automating the mathematical analysis of our approach. It integrates the two files generated by the first block and also the features reference files. Thereby, it makes a comparison of the evolution histories of the two population and generate a log file which contains the traces of the comparison and the resulted differences.

6.3 Perfect Co-evolution Analyzer

The third block deals with the perfect co-evolution restoration, it integrates the log file generated by the second blocks and displays the features that may be restored to the platform. this block gives also the possibility to integrate the two matrices with the features reference files and applies the perfect co-evolution restoration algorithm which we previously elaborated and generate a new matrix for the SPL platform.

Firstly we started by developing the first block, currently we are performing a list of test cases to validate it. Subsequently, we intend after that to develop the second and the third functional blocks. We envisage after finalizing the three blocks, integrating them into an application for complete co-evolution analysis.

7 CONCLUSIONS

Our general research work addresses the problem of change propagation in evolving software product lines. Thereby, we deal with the analysis of co-evolution of the platform and the products of the software product line to understand how they impact each other during their evolution. In this paper, we have first proposed the use of biological technique cladistics to illustrate the evolution history of the products and the platform through evolutionary trees, each tree represents the history of a product, then we established a mathematical analysis to compare the trees and to find out the changes of product that were not propagated to the platform and finally we elaborated an algorithm to help repropagating the missing features to the platform. We illustrated the approach

through a case study on the mobile media software product line. There are several perspectives to our current work. Firstly, we will develop a tool to automate the co-evolution analysis, we separated the tool to three main functional blocks representing the three steps of the approach. The first block is currently developed and under validation. Secondly, we aim to improve the third step of our approach to act directly on the feature models of the platform and correct them.

REFERENCES

- Ajila, S. A. and Kaba, A. B. (2008). Evolution support mechanisms for software product line process. *Journal of Systems and Software*, 81(10):1784–1801.
- Anderson, R. M. and M. May, R. (1982). Coevolution of hosts and parasites. *Parasitology*, 85(02):411–426.
- Anquetil, N., Kulesza, U., Mitschke, R., Moreira, A., Royer, J., Rummler, A., and Sousa, A. (2010). A model-driven traceability framework for software product lines. *Software and Systems Modeling*, 9:427–451.
- Benlarabi, A., A. Khtira, A., and Asri, B. E. (2014a). An analysis of domain and application engineering co-evolution for software product lines based on cladistics: A case study. In *ICSEA'14 Forthcoming (2014)*, pages 495–501.
- Benlarabi, A., Khtira, A., and Asri, B. E. (2014b). A co-evolution model for software product lines: an approach based on evolutionary trees. In *WCCS14. IEEE Forthcoming (2014)*.
- Brinkman, Fiona, S., and Leipe, D. D. (2004). *Bioinformatics: a Practical Guide to the Analysis of Genes and Proteins*, volume 43. John Wiley & Sons.
- Clements, P., Northrop, L., and Boehm, B. W. (2002). *Software Product Lines : Practices and Patterns*. Fondo Xavier Clavigero, S.J. ITESO.
- Ehrlich, P. R. and Raven, P. H. (1964). Butterflies and plants: a study in coevolution evolution. *JSTOR*, pages 586–608.
- Fitch, W. and Margoliash, M. (1967). Construction of phylogenetic trees. *Science*, 155(760):279–284.
- Goknil, A., Kurtev, I., van den Berg, K., and Veldhuis, J. (2011). Semantics of trace relations in requirements models for consistency checking and inferencing. *Software & Systems Modeling*, 10:31–54.
- Kastner, C., Thum, T., Saake, G., FeigenspanLeich, J., Wielgorz, T. F., and Apel, S. (2009). Featureide: A tool framework for feature-oriented software development. In *IEEE 31st International Conference on Software Engineering*, pages 611–614. IEEE.
- Pohl, K., Bckle, G., and van der Linden, F. J. (2005). *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer.
- Romero, D., Urli, S., Quinton, C., Blay-Fornarino, M., Collet, P., Duchien, L., and Mosser, S. (2013). Splemma:

A generic framework for controlled-evolution of software product lines. In *The 17th International Software Product Line Conference*, pages 59–66.

Tizzei, L. P., Dias, M., Rubira, C. M., Garcia, A., and Lee, J. (2011). Components meet aspects: Assessing design stability of a software product line. *Information and Software Technology*, 53(2):121–136.

