Information **2015**, *6*, 550-563; doi:10.3390/info6030550

OPEN ACCESS

information

ISSN 2078-2489

www.mdpi.com/journal/information

Article

Analyzing Trends in Software Product Lines Evolution Using a Cladistics Based Approach

Anissa Benlarabi *, Amal Khtira and Bouchra El Asri

IMS Team, Laboratory SIME, ENSIAS, University Mohamed V, Rabat 10500, Morocco;
E-Mails: amalkhtira@gmail.com (A.K.); elasri@ensias.ma (B.E.A.)

* Author to whom correspondence should be addressed; E-Mail: a.benlarabi@gmail.com;
Tel.: +21-26-6841-4995.

Academic Editor: Ahmed El Oualkadi

Received: 16 June 2015 / Accepted: 21 August 2015 / Published: 27 August 2015

Abstract: A software product line is a complex system the aim of which is to provide a platform dedicated to large reuse. It necessitates a great investment. Thus, its ability to cope with customers' ever-changing requirements is among its key success factors. Great effort has been made to deal with the software product line evolution. In our previous works, we carried out a classification of these works to provide an overview of the used techniques. We also identified the following key challenges of software product lines evolution: the ability to predict future changes, the ability to define the impact of a change easily and the improvement in understanding the change. We have already tackled the second and the third challenges. The objective of this paper is to deal with the first challenge. We use the cladistics classification which was used in biology to understand the evolution of organisms sharing the same ancestor and their process of descent at the aim of predicting their future changes. By analogy, we consider a population of applications for media management on mobile devices derived from the same platform and we use cladistics to construct their evolutionary tree. We conducted an analysis to show how to identify the evolution trends of the case study products and to predict future changes.

Keywords: software product lines; evolution; cladistics

1. Introduction

Software product line engineering consists of building a common platform for a set of products dedicated for a specific business domain [1]. The main advantage of software product line engineering is the improvement of the productivity [2]. By using the software product line (SPL) paradigm, customers get products adapted to their needs and wishes at a reasonable price and a short time.

However, customers' requirements evolve continuously. Thus, the SPL must be adapted to cope with new customers' needs. Compared to single software evolution, SPLs evolution remains more difficult because the components of the platform are shared by many products. Hence, the change of a component has a large impact. In order to deal efficiently with SPL evolution, we identified the three following challenges: the ability to plan the evolution and to predict future changes, the ability to define the impact of a change easily, and the improvement in understanding the change. In our previous work, we separated works done to deal with SPL evolution into three groups: traceability approaches, evolution modeling approaches and assessment techniques [3]. These approaches focus mainly on the second and the third challenges. In ordcer to tackle the three challenges, we proposed a co-evolution approach for SPLs and also a design for a co-evolution analysis framework based on biological techniques. We discussed in detail the second and the third challenges previously [3] while the first challenge was less developed.

In this paper, we present an approach to model the evolution of the derived products of a SPL in order to understand the history of the SPL and to interpret the process of descent of its products. We use the cladistics biological technique which was extensively used in biology to understand the evolution of a set of organisms offspring from the same ancestor and to predict their evolution trends [4]. By analogy with biology, cladistics studies the evolution of a set of organisms derived from the same ancestor by drawing an evolutionary tree of them. In SPL, the organisms are the products derived from the common platform. Hence, our work consists of building the evolutionary tree of a set of products derived from an SPL, and conducting analysis of this tree to show how this approach can enhance the evolution understanding and help predicting the future changes, identifying the evolution trends and also improving the design decision.

The remainder of this paper is divided into five sections. In Section 2, we present challenges related to SPL evolution, then we remind the results of our previous work and related works. In Section 3, we present our motivation behind using biological techniques to study SPL evolution. Section 4 illustrates our approach through a case study on the mobile media SPL. In Section 5, we discuss the results of our approach and we present further steps. We give a conclusion in Section 6.

2. Software Product Lines Evolution Challenges

In this section, we present the SPL paradigm and the SPL evolution problem. Then, we extract our research questions and we introduce our approach.

2.1. Software Product Line Engineering

The SPL engineering consists of building a common platform for a set of products dedicated to a specific business domain. The SPL can be built from scratch by studying all the domain requirements, or by re-engineering the existing products [5]. The platform is formed by common domain assets that are developed by taking into consideration the commonality and the variability of the domain applications [1]. The SPL engineering encompasses three processes, the domain engineering process in which the common platform is developed, the application engineering process in which the applications of the family are derived from the common platform, and the management which includes the management of the two previous processes and also the management of business needs evolution [2]. The main advantage of SPL engineering is the improvement of the productivity. By reusing the domain assets [2], customers will get customized products adapted to their needs and wishes at a reasonable price and a short time. However, in our rapidly changing world, business strategies continuously evolve. Hence, information systems must be able to cope with business strategies evolution. This point led us to focus on software evolution especially SPLs evolution, because it is more challenging than single software evolution and presents a sensitive issue.

2.2. Separating Software Product Lines Evolution from Software Evolution

Software evolution is an important activity because it must not only preserve the system and fix the faults but it must also accompany the evolution of business requirements [6]. Thus, a great interest must be given to the evolution activity in order to preserve the software for a long time and to protect it from the aging phenomenon [7]. Compared to single software evolution, the SPL evolution is more complicated because the SPL changes impact two levels: The level of products and the level of the common platform. Indeed, the requirements changes must be propagated in the two levels, which requires a well understanding of the change and a precise determination of its impact.

Moreover, the main objective of a SPL is the large scale reuse, and in order to achieve this goal, the SPL developers must be able to anticipate the business needs to be able to derive future products. In order to deal efficiently with the SPL evolution, we introduced the following challenges [3]:

- The ability to plan the evolution and to predict future changes
- The ability to define the impact of a change easily
- The improvement of the change understanding

2.3. Our Previous Work

In our previous work, we classified works that tackle the SPL evolution into three groups [3]. The first group is traceability approaches which capitalize on knowledge about traceability of links between the domain assets and the links between the domain and the application assets and give a rich knowledge base for change impact analysis. The second group is the modelling approaches that deal with the implementation of change and give a set of rigorous steps for controlled evolution. The final group concerns the techniques of assessment that validate the change even before its implementation, which allows for anticipating defects and the undesirable behaviours. We mention also the work of

Svahnberg [8] who presented a set of guidelines for SPL evolution with categorizations of the SPL changes in order to improve the SPL change understanding. These works focused just on the second and the third challenges and they neglect the changes triggered by application engineering.

In order to tackle the three challenges, we proposed an approach based on cladistics [4], which is a biological technique used to understand how organisms evolve over time. It builds an evolutionary tree for a population by classifying its members on the basis of the evolution of their physical characters or the evolution of their behaviour. Thus, we firstly collected the evolution history of the SPL and each derived product embodied in their feature models. Then we used cladistics to construct the evolutionary trees of the platform and the derived products. Thus, we compared the resulted trees, and finally we applied the mathematical analysis that we elaborated whose aim is the restoration of the perfect co-evolution of the platform and its products [9].

Our approach tackles the three challenges. It allows for defining the impact of the change by identifying the hidden links between the domain and the application assets. It also improves the change understanding through a synthesis of the history of the SPL evolution and helps predicting future changes by considering changes that were implemented at the products level and may trigger the evolution of the platform. The second and the third challenges were discussed in details in our previous works [10]. However, the first challenge was less developed. The objective of the research reported in this paper is to develop the first challenge, and to show how our approach for modeling evolution of SPLs, can capture changes happening to the products, and then help deducing evolution trends of the SPL and predicting its future changes.

2.4. Related Works

Research about software evolution is being continuously in progress. A number of recent researches have focused in analyzing software evolution. In this section, we mention some of them.

Lin [11] proposed an approach to understand and simulate a software evolution based on its source code by extracting the historical changes made to the project classes. A source code change is associated with the change type given according to a taxonomy and the changed entity. These historical data are then used to calculate the frequency of each change type, and to deduce the probability of a change made on each entity. The evolution simulations will generate future changes for each entity based on this probability and then apply those changes.

Bhattacharya [12] presented a graph based analysis for software evolution. The approach consists of building two types of graphs, the source code-based graph which captures communication between the program functions and modules, and the developer collaboration graph which shows how developers communicate as software evolves. Based on these graphs, a set of metrics are calculated. Through a case study on eleven open source projects, the authors substantiated the ability to predict attributes such as bug severity, effort, and defect count by examining the evolution of the calculated metrics over time.

Kanda [13] focused on SPLs. He proposed an approach to extract the evolution history of a domain software products from their source code in order to help the developers of this domain in the products re-engineering. The output of this approach is a products evolution tree. The nodes of the tree represent the products, the edges connect similar products, and each edge label indicates the similarity of products

that connects. The tree is constructed through two steps: calculation of file-to-file similarity for all source file pairs of all products, the construction of minimum-spanning tree of products. The cost of an edge connecting two products is calculated based on the number of similar files between the products. The approach was validated through a case study on the PostgreSQL tool. The results show that 53% to 92% of edges were consistent with the actual evolution history.

These approaches focus on technical evolution, functional evolution has not yet been covered extensively despite the importance of the functional analysis in finding solutions to the business needs. Unlike the presented works, our approach focuses on the evolution of feature models of the software as it represents the business view. Our goal is to understand the business trends through the functional analysis of the software evolution. In our work we focus on SPLs evolution.

3. Modeling Evolution of SPLs through Cladistics

In this section, we present the cladistics classification approach with an example from biology, then we present our motivations to introduce this technique in SPLs.

3.1. Biological Evolution vs. Software Evolution

In biology, the evolution is not a simple term describing the changes happening to organisms. It is defined as a genetic change in a population that is inherited over several generations [14]. The central idea of biological evolution is that all organisms on earth are classified in populations [15]. Each population has a common ancestor from which all its organisms are descended. The common ancestor gave rise to the diversity of the population by reproduction.

Hence, biological evolution helps in understanding the history of life and provides a basis for interpreting the relationships between the organisms of a population and the process of descent. Without the theory of evolution, we will not find satisfactory explanation for life history strategies.

In our previous work, we illustrated how the evolution is the result of interaction between organisms in nature. We discussed the co-evolution which is the examination of patterns of interaction between major groups of organisms with a close and evident ecological relationships, such as plants and herbivores [16]. Hence, evolution is an evident result of the coexistence of many species. This conclusion does not concern only the nature but also the other fields such as SPL engineering where the change is driven by the continuous need to adapt the platform and the products to new business needs.

The objective of this paper is to model the evolution of the derived products of a SPL, in order to understand the history of the SPL and to interpret the process of descent of the family products. These results can yield a significant conclusions about the evolution trends of the SPL, and its business field.

3.2. Modeling Software Product Line Evolution Using Cladistics

In biology, the evolution of a population of organisms offspring from the same ancestor is modelled though their phylogenetic tree [17]. A phylogenetic or evolutionary tree of a population is a branching diagram or tree 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 methods to construct the phylogenetic trees, the phenetics method and the cladistics method [17]. The first is a distance-based method that measures differences among DNA sequences of the members of a population and builds the tree totally from resultant distance matrix, while the cladistics method is a character-based method. We choose the cladistics method because it assumes that a set of organisms are descended from a common ancestor by mutation of its physical or behavioural characters. Similarly to biology, the characters of the products are their features. The method consists in the following steps:

(1) Organisms selection: The organisms of the classification should be necessarily related by ancestor-descendant relationships or indirect relationships based on common ancestry.
(2) Characters extraction: The characters include physical and behavioural characters that allow to make distinction between different organisms, e.g., the wings and the feathers are two characters for birds. Each character has two or more exclusive states, e.g., its presence or absence in an organism denotes two different states, the absence of the character is called the "primitive" state and its presence is called the "derived" state.
(3) Characters matrix construction: This matrix summarizes the results of organisms selection and characters extraction. The organisms are put in lines and the characters are put in columns. Then, the matrix is filled by numbers that represents the codification of the states of the characters.
(4) Cladogram drawing: The cladogram is a tree constructed on the basis of the characters matrix by grouping organisms together based on their shared characters. All the organisms are in the endpoints of the tree. Each node regroups shared characters between all the next organisms unless these characters are modified later. The order of characters in the tree is determined by its number of occurrence in the matrix.

Such classification will represent the evolution of users' needs, and then, it can enhance the understanding of the business strategy. It is also based on a comparative study and gives a knowledge base on the SPL evolution history. Cladistics method produces hypothesizes about the relationships of organisms in a way that predicts properties of the organisms. Unlike previous systems of analyzing relationships, cladistics is explicitly evolutionary. Hence, it is possible to examine the way in which characters change within groups over time, the direction in which characters change, and the relative frequency with which they change. In the next section, we present our approach through a case study.

4. Experimental Section

In this section, we use the cladistics classification in order to understand the process of descent in the SPL and also to provide a basis for interpreting the relationships among the applications and their evolution trends. Firstly, we present generally the steps of our proposed approach and then we illustrate each step through an example.

4.1. Our Approach

Our approach is illustrated in Figure 1. It contains four main steps, the first consists in features extraction. In the second step, we build the features matrix, then we draw the evolutionary tree of the

population in step 3. And finally, we perform an analysis of the tree to show how to enhance the change understanding through our approach.

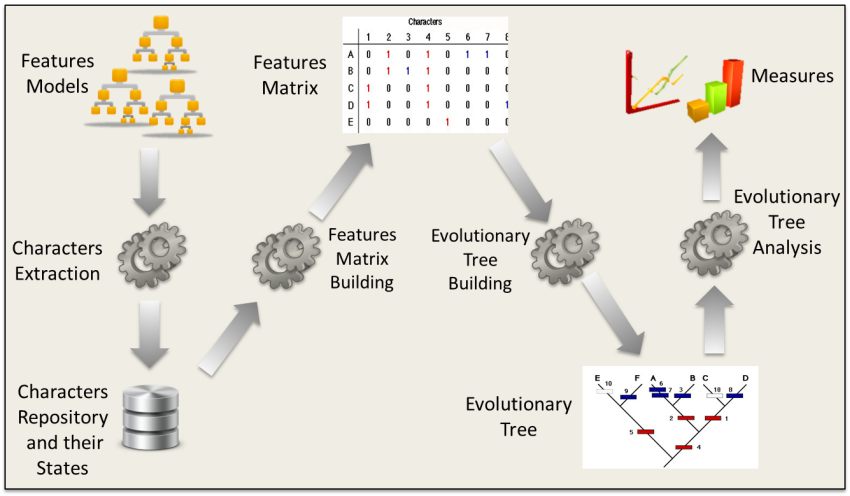

Figure 1. Cladistics classification for software product line (SPL).

4.2. Population Selection

We investigate the evolution history of a SPL. We consider a set of populations formed by the products of the SPL. These products share the same ancestor which is the common platform of the SPL. The products are generated by descent with modification. The target SPL for our case study is a public SPL that was developed with two technologies, the aspect oriented programming and the object oriented programming, to be a useful case study in the benchmarking of the two technologies. It manipulates photo, music, and video on mobile devices such as mobile phones [18]. It has more than 200 derived products. Many evolution scenarios were performed on the mobile media SPL. Thus, we have many products of the SPL. Our population is formed by 11 products of the mobile media SPL. Table 1 gives a description of the population and lists in details the features of each product extracted from its feature model [19].

4.3. Characters Extraction

The characters are the physical or the behavioural traits that distinguish an organism. By analogy, in software engineering, the characters of a software application are its visible traits that can only be its features. For the populations of the SPL products, we extracted 19 characters from the description of the table below that shows the difference between the products. Some characters has two states, the state 0 which denotes its absence and the state 1 which denotes its existence. Other characters have three states such as creating media which can be photo or video or music. The tree states are encoded as three binary independent characters. Table 2 identifies the characters and their states encoding. We gathered together the characters that share the same states.

Table 1. Mobile Media SPL Population.

Population	Description
Product 1	An application for photo management. It encompasses the following features: Create album, delete album, create photo, delete photo, view photo, sort photo, capture photo, and edit photo label
Product 2	An application for photo management. It encompasses the same features as the product 1 and two new features: Set favourites and see favourites
Product 3	An application for photo management. It encompasses the same features as the product 2, the possibility to create many copies of the photo was added
Product 4	An application for photo management. It encompasses the same features as the product 2. Other features related to sending and receiving Photos were added
Product 5	An application for music management. It encompasses the following features: Create album, delete album, create voice, delete song, play song, sort song, and edit song label
Product 6	An application for music management. It encompasses the same features as the product 5 with the possibility to set and see favourites
Product 7	An application for music management. It encompasses the same features as the product 6 with the possibility send and receive song
Product 8	An application for video management. It encompasses the following features: Create album, delete album, create video, delete video, play video, sort video, and edit video label
Product 9	An application for video management. It encompasses the same features as the product 8 with the possibility to set and see favourites
Product 10	An application for video management. It encompasses the same features as the product 9 with the possibility to send video and receive video
Product 11	An application for photo management. It encompasses the same features as the product 4 with the possibility to share photo in social application and modify photo (trim and resize)

Table 2. Characters of the population and their states.

Character	Code	State
1 Create album	0	Absence
	1	Existence
2 Delete album	0	Absence
	1	Existence
3 Create Media		
4 Delete Media		
5 View Media	000	Absence
6 Edit Media Label	001	Type of media is photo
7 Sort Media	010	Type of media is photo or music
8 Set Favorites	100	Type of media is photo or music
9 View Favorites		or video
10 Copy Media		
11 Send Media		
12 Receive Media		
13 Capture Photo		
14 Capture Video		
15 View Photo	0	Absence
16 Play Song	1	Existence
17 Play Video		
18 Share Photo		
19 Modify Photo		

4.4. Features Matrix Construction

The characters matrix summarizes the states of characters for each organism of the population. Hence, we represent the products of the SPL with their characters. Table 3 represents this characters state matrix. Each line of the matrix represents the states of the nineteen characters in a product. For the characters that support two states, 0 denotes its absence from the product and 1 its existence. For the characters that support three states, it depends on the variant type of media. 000 denotes its absence, 001 indicates that the character exists but with the variant photo, 010 and 001 indicate that the character exists respectively with the variants music or video.

Table 3. Matrix of characters states.

Variants	**Characters**																		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
P1	1	1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	1	0	0	0	0	0	0
P2	1	1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 0	0 0 0	0 0 0	1	0	0	0	0	0	0
P3	1	1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 0	0 0 0	1	0	0	0	0	0	0
P4	1	1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	1	0	0	0	0	0	0
P5	1	1	0 1 0	0 1 0	0 1 0	0 1 0	0 1 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0	0	0	1	0	0	0
P6	1	1	0 1 0	0 1 0	0 1 0	0 1 0	0 1 0	0 1 0	0 1 0	0 0 0	0 0 0	0 0 0	0	0	0	1	0	0	0
P7	1	1	0 1 0	0 1 0	0 1 0	0 1 0	0 1 0	0 1 0	0 1 0	0 0 0	0 1 0	0 1 0	0	0	0	1	0	0	0
P8	1	1	1 0 0	1 0 0	1 0 0	1 0 0	1 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0	1	0	0	1	0	0
P9	1	1	1 0 0	1 0 0	1 0 0	1 0 0	1 0 0	1 0 0	1 0 0	0 0 0	0 0 0	0 0 0	0	1	0	0	1	0	0
P10	1	1	1 0 0	1 0 0	1 0 0	1 0 0	1 0 0	1 0 0	1 0 0	0 0 0	1 0 0	1 0 0	0	1	0	0	1	0	0
P11	1	1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	0 0 1	1	0	0	0	0	1	1

4.5. Cladogram Drawing

Cladogram is a diagram which represents the relationships between different groups of organisms. Cladogram reconstructs the evolutionary history of the organisms. Cladograms can also be called evolutionary trees. The cladogram is constructed by grouping organisms together based on their shared derived characters. Figure 2 represents the cladogram of the selected mobile media applications. the characters of Table 2 are represented by the set $Ciwhere 0 < i \leq 19$. The characters that have three states (photo, music or video) are represented as follows $Ci.jwhere 0 < i \leq 19, 0 < i \leq 3$.

The first node includes the characters that are shared between all the organisms. They are deduced from the matrix because they have the state 1 for all the products. The tree is then separated to many edges, the next nodes as the first one, gathers the common characters between the descended products, these characters are extracted from the matrix by examining their states for the products.

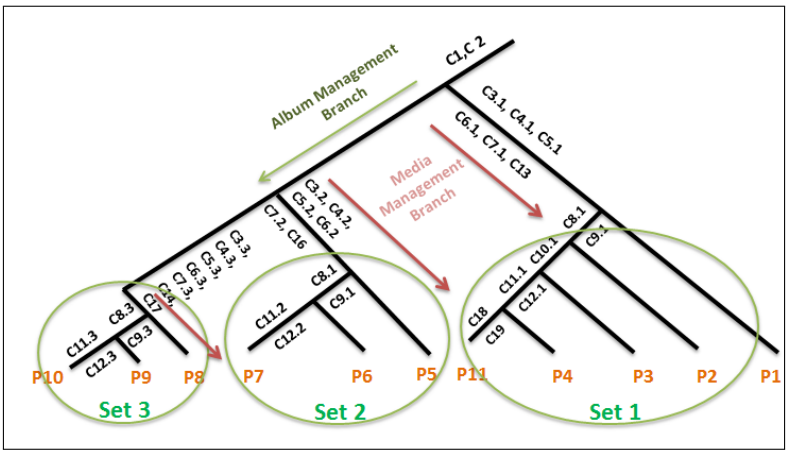

Figure 2. Cladogram analysis.

5. Results and Discussion

The mobile media case study yielded a cladogram of 11 leafs representing each one a mobile media application. The cladogram can be divided into three sets 1, 2 and 3. Each set gathers the similar variants, and represents a subfamily of products that share common variants. In addition, the cladogram shows the occurrence of the variants. Thus, we can deduce the most frequent variants in the mobile media applications and also the less frequent ones. In this section, we conduct an analysis of evolution trends based on the graphical view of the cladogram produced by the cladistics classification. We provide also a set of measures that help interpreting the cladogram.

5.1. Graphic Based Analysis

In the graphical analysis, we explore the splitting scheme and the branches of the cladogram in order to deduce many results concerning the evolution trends, the design decision and the future changes.

5.1.1. Identifying Evolution Trends

The evolution trends identified by using cladograms can prompt designers to contemplate new design possibilities and alternatives for future products. By examining the cladogram in Figure 2, specifically the splitting scheme at each branch, it can be noticed that splitting leads to three main sets, where set 3 contains most of the splitting action and includes the greatest share of variants. This indicates a preference for future evolution trends in the direction of this set. The set 3 contains features related to media management. Hence, the evolution of the mobile media applications concerns mostly the media management features. The branch of albums management was not evolved since the creation of the first product. However, it is included in all the products. This brings us to the question of priority of features in such applications. Since the media management features are the core features, they are concerned by the improvement and the evolution more than secondary features.

5.1.2. Improving Design Decision

The most evolution scenarios happened to the branch of media management, while album management are still in the primitive state since the creation of the first product, two hypothesis can be made. First, customers are less interested in the album management, the creation and the deletion are sufficient to manage albums. Second, developers did not think to give more features in albums management, such as sorting albums and set favourites albums. Such observation can help designers making decisions about the future of the branch, and also decisions about the need of evolving it or creating new features concerning album management.

5.1.3. Anticipating Future Changes

The cladogram is separated into three sets. The first set contains the largest number of nodes and the maximum of variants, which reveals that there is a preference for future evolution in this set. We notice also that the evolution scenarios that happened for the photos application are transmitted after to the videos applications and songs applications. Hence, we can anticipate the future changes, by developing variants of the new features of the photos application for the videos and songs applications. By considering the cladogram in Figure 1, new variants for videos and songs can be deduced from the features share photo and modify photo of the photo application 12. The possibility to share a media is common between the three types of media, while the possibility to modify can be customized for each type of media. For photos, the modification encompasses the resizing and the trimming. For a video or a song, the modification can allow for cutting a portion of the video or the song.

5.2. Measures Based Analysis

We introduce a set of measures that we calculate on the cladogram. Apart from understanding and predicting changes, these measures can help in identifying most important features and the degree of similarity between products.

5.2.1. Cladogram Metrics

We propose the following the measures to analyze the cladogram:

- Feature Weight: To measure the importance of features, we calculate for each feature its weight which represents the number of occurrences of the feature in the population.
- Node Range: This measure is calculated for each node in order to find out the characters that are shared between the maximum of SPL applications. It represents the number of leafs descended from the node.
- Evolution Distance: It measures the number of changes that separate two leafs. By calculating the average distance, we can estimate the difference between the SPL products.

We calculated these measures for the cladogram of our case study. In Figure 3, we present the resulted calculations. By investigating the graphics, we can deduce many results:

- The feature weight graphic shows the degree of sharing of features and also their importance. Therefore, it helps identifying the impact of a feature change. This measure can also help in deriving new products. Features that have the maximum weight are the mandatory ones, and also the features that customers have a preference for. The graph can be much more useful for SPLs with substantial number of features. It gives a quick and real view on the use of features.
- The evolution distance graphic presents the differences between the products. It can also reveal the degree of similarity between two products. The value of this metric for two successive products increases when the difference between them is considerable. Similarity between successive products can enhance our understanding for customers preferences. Low value for this metric proves weak diversity among the SPL products.
- The node range graphic shows the importance of each node and also the tendency to derive products from a node. A node with high number of derived products gathers the features shared by the majority of products. Otherwise, the possibility for new derived product to be under this node is of great importance.

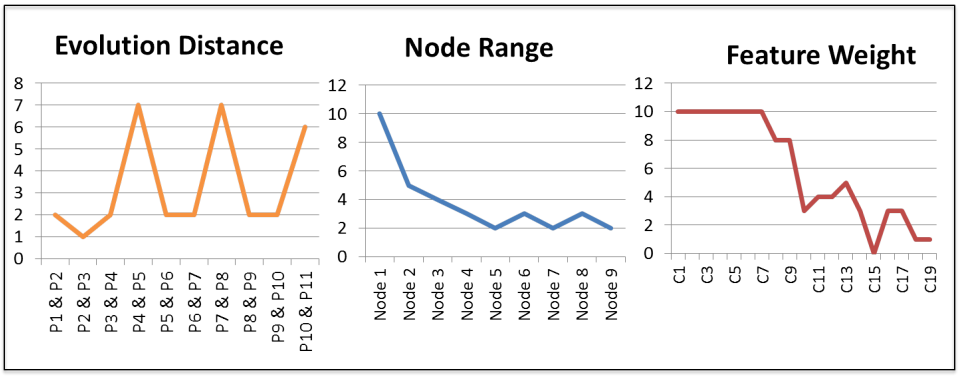

Figure 3. Measures calculation for mobile media cladogram.

6. Conclusions and Perspectives

In this paper, we introduced a cladistics based approach to study the evolution of SPLs, which consists in building the evolutionary tree of a set of applications derived from a SPL based on their shared features. Our purpose is to investigate the evolution trends of the SPL in order to study the ability to predict future changes. By investigating the frequency of splitting in the tree, we can identify the features that evolve mostly, and thus, we deduce the preference of customers. In addition, the examination of the branches of the tree can yield a significant hypotheses that help developers make design decisions.

We applied our proposal on the mobile media SPL. The population of the study is formed by 11 products that manipulate different media: Photo, music, and video on mobile devices such as mobile phones. We constructed the evolutionary tree of the products using the cladistics classification and we discussed the resulted tree to show how this approach can help in anticipating future changes and also in enhancing the change understanding.

We conducted our case study as well as the metrics calculation and analysis manually. In further steps, we focus substantially on validating our approach through other case studies and also on implementing a framework to automate our approach. We already selected tools for feature modelling and cladogram

drawing. Currently, we are in the first step of our automation project. The aim of this step is to produce a cladogram for a SPL based on feature models of its products modelled with the selected feature modelling tool. After achieving this step, we intend to realize a program for calculating our proposed metrics and displaying their graphs.

Author Contributions

Anissa Benlarabi wrote the paper. Anissa Benlarabi and Amal Khtira designed and performed the experiments. Anissa Benlarabi and Bouchra El Asri analyzed the resulted data. All authors have read and approved the final manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Pohl, K.; Böckle, G.; van der Linden, F.J. *Software Product Line Engineering: Foundations, Principles and Techniques*; Springer-Verlag: Berlin, Germany, 2005.
2. Clements, P.; Northrop, L.; Boehm, B.W. *Software Product Lines : Practices and Patterns*; Addison-Wesley: Boston, MA, USA, 2002.
3. Benlarabi, A.; Khtira, A.; El Asri, B. A co-evolution model for software product lines: An approach based on evolutionary trees. In Proceedings of the Second World Conference on Complex Systems, Agadir, Morocco, 10–12 November 2014.
4. Brinkman, F.S.L.; Leipe, D.D. *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*; John Wiley & Sons: New York, NY, USA, 2004; Volume 43.
5. Ziadi, T.; Henard, C.; Papadakis, M.; Ziane, M.; Le Traon, Y. Towards a language-independent approach for reverse-engineering of software product lines. In Proceedings of the 29th Annual ACM Symposium on Applied Computing, Gyeongju, Korea, 24–28 March 2014; pp. 1064–1071.
6. Bennett, K.H.; Rajlich, V.T. Software maintenance and evolution: A roadmap. In Proceedings of the Conference on the Future of Software Engineering, Limerick, Ireland, 4–11 June 2000; pp. 73–87.
7. Parnas, D.L. Software agingm. In Proceeding of the 16th International Conference on Software Engineering, Sorento, Italy, 16–21 May 1994; pp. 279–287.
8. Svahnberg, M.; Bosch, J. Evolution in software product lines. *J. Softw. Maint. Res. Pract.* **1999**, *11*, 391–422.
9. Benlarabi, A.; Khtira, A.; El Asri, B. Co-evolution analysis for software product lines. In Proceedings of the 10th International Conference on Evaluation of Novel Approaches to Software Engineering, Barcelone, Spain, 29–30 April 2015.
10. Benlarabi, A.; Khtira, A.; El Asri, B. An analysis of domain and application engineering co-evolution for software product lines based on cladistics: A case study. In Proceedings of the Ninth International Conference on Software Engineering Advances, Nice, France, 12–16 October 2014; pp. 495–501.

11. Lin, Z. Understanding and simulating software evolution. In Proceedings of the International Conference on Software Engineering, San Francisco, CA, USA, 18–26 May 2013; pp. 1411–1414.
12. Bhattacharya, P.; Iliofotou, M.; Neamtiu, I.; Faloutsos, M. Graph-based analysis and prediction for software evolution. In Proceedings of the 34th International Conference on Software Engineering, Zurich, Switzerland, 2–9 June 2012; pp. 419–429.
13. Kanda, T.; Ishio, T.; Inoue, K. Extraction of product evolution tree from source code of product variants. In Proceedings of the 17th International Software Product Line Conference, Tokyo, Japan, 26–30 August 2013; pp. 141–150.
14. Futuyma, D.J. *Evolutionary Biology*, 3rd ed.; Sinauer Associates Inc.: Sunderland, MA, USA, 1998; Volume 4.
15. Mayr, E.; Bock, W.J. Classifications and other ordering systems. *J. Zool. Syst. Evolut. Res.* **2002**, *4*, 169–194.
16. Rosenthal, G.A.; Berenbaum, M.R. *Herbivores Their Interactions with Secondary Plant Metabolites: Ecological and Evolutionary Processes*; Academic Press: Waltham, MA, USA, 1992; Volume 2.
17. Fitch, W.M.; Margoliash, M. Construction of phylogenetic trees. *Science* **1967**, *155*, 279–284.
18. Tizzei, L.P.; Dias, M.; Rubira, C.M.; Garcia, A.; Lee, J. Components meet aspects: Assessing design stability of a software product line. *Inf. Softw. Technol.* **2011**, *53*, 121–136.
19. Kang, K.C.; Kim, S.; Lee, J.; Kim, K.; Shin, E.; Huh, M. FORM: A feature-oriented reuse method with domain-specific reference architectures. *Ann. Softw. Eng.* **1998**, *5*, 143–168.