A Novel Approach for Feature Selection Support of a Software Product Line Development

Pujianto Yugopuspito[†,††], I Made Murwantara[†,††], and Adrian Hartanto Sutomo[†]

[†]Informatics Dept., Faculty of Computer Science, [††] Research and Development Computer Lab.
Universitas Pelita Harapan, Banten, 15811 Indonesia

Summary
Features selection of a feature model is the key point of a Software Product Line development. It selects the most suitable configuration of features that will be included into a Software Product Line, based on the user requirements. However, features of a feature model may comprise dependency and multi functionality that correspond to the quality attributes, where it will pin points the specific product. Further, features may also have constraints to their relationship, which will influence the whole elements of product configuration. To this, we proposed an approach that utilizes the hybrid Analytic Network Process (ANP). In our proposed approach, first, the groups of features functionality are grouped to identify the non-functional factor. Then, the non-functional factors are grouped in a network for a specific quality attribute. Subsequently, the networks of specific quality attributes are merged to have global view of quality attribute dependency. At the end, the dependency for each group of quality attributes is calculated using AHP and ANP. The final result is composition guidance for product line configuration. In this paper, we use an eLearning Product Line (eLPL) case study to demonstrate the feasibility of our approach.

Keywords:
Software Product Line Engineering, Feature, Feature Model, Analytical Network Process, Analytical Hierarchy Process.

1. Introduction

Software Product Line Engineering (SPLE) has its objective to produce specific software product, from the product's member that are included into a product family, with high quality, low cost and within short time [1]. SPLE, heavily, uses *reuse* paradigm to achieve its goals. SPLE consists of two large processes that are Domain Engineering and Application Engineering. In the Domain Engineering, the abstraction of member products (including requirement document) are configured to have specific product. Meanwhile, the Application Engineering configures the architecture of specific product that derives from the Domain Engineering process [2].

In SPLE, feature modeling [3] is a requirement engineering approach that suitable to represent the commonality and variability of a product line development. Where, a feature model abstracts the software system requirements or characteristics of features that both customer and developers can understand.

In feature model, there are two types of feature that are common and variable features. The common feature will be included into all member products, because it exists in all products. On the other hand, the variable features must be selected to have specific software product.

The feature grouping is a method to get specific information, which corresponds to the functional and non-functional. Further, groups of functional may end up as the quality attributes. In this case, if a feature or functionality in a quality attribute is deceased, then the quality attribute will change its characteristics or behavior.

The feature selection in a Software Product Line (SPL) development is becoming complex because of dependency between features of feature model. Features of feature model are structured in a hierarchy model. Meanwhile, the dependency between features may include hierarchy in a different tree in the same model. Furthermore, dependency of quality attributes may affect the final configuration. The crosscut between features makes it hard to configure a SPL in terms of quality attribute. Similarly, the concern that crosscut may increase the complexity of SPL configuration.

This paper investigated the support for selection of the features of feature model in a configuration for specific product. The quality attributes dependency that is based on the cluster of non-functional factors. The dependency management may influence the decision of selection. Further, efficient configuration strategy may reduce the complexity of quality attributes dependency within a product family. It aim to capture and evolve a SPL's assets so as to gain insight into features diversity, efficiently. Where, the quality attributes are the overall factors that influence the system under configuration.

The Analytical Network Process (ANP) according to Saaty [4] *"is a general theory of relative measurement used to derive composite priority ratio scales from individual ratio scales that represent relative*

Manuscript received June 5, 2012
Manuscript revised June 20, 2012

measurements of the influence elements that interact with respect to control criteria". The ANP uses a pair wise comparison to measure the weight of its elements structure, to rank the alternatives of decision. In the ANP, it taking into accounts the dependency between elements through network, where the network establish the relationship of ANP's elements. Using the ANP, the features of feature model are grouped to form the non-functional factors. The non-functional factors are also grouped to have specific quality attributes. Clustering of quality attributes may comprise redundant features. The elements of quality attribute may have relation to others that forms the dependency between quality attributes. In this case, if a quality attribute is included into a SPL configuration then the dependency should be considered to be included. Otherwise, the goal of SPL development cannot be achieved.

The quality attribute is formed from a group of functionality. In the feature model, features have functionality. So, if sets of features are grouped, then it will form a quality attribute. Furthermore, feature model comprises features, which may be grouped for specific goal. And, it will establish lots of quality attribute.

Features of feature model may have dependency between them because of the requirements. In this case, the dependency between features is the constraints of the product configuration in the SPLE development. Using ANP, the dependency can be modeled, and the decision of dependency may be explicated straightforward during product configuration process.

In this paper, we propose the novel way to manage the dependency between features in terms of quality attributes. Our research focuses on how to provide support for software architecture to select the features of feature model, in order to have efficient and less dependency conflicts during SPLE configuration. In the next part, we will show related work. And, it is followed by the research methodology. We also provide a case study on eLearning Product Line (eLPL) to show the reliability of our method. Finally, the last section will conclude our paper.

2. Contribution

The contribution of this work is on the use of multi scale approach to have a better selection method in a SPL development. The cluster of features forms the quality attributes which eventually will also explicit the features dependency that affect the quality attributes relationship.

To manage the features dependency, we propose to use the ANP. The ANP is a combined multi scale and network approach, which we found match to our goal.

3. Related Work

Supporting the selection process on features of feature model that focuses on features dependency already been done. In FORMS [5], features are groups into functional and non-functional. Features are refined across layers to have better information of non-functional factor. Its layers are the Capability Layer, the Operating Environment Layer, the Domain Technology Layer and the Implementation Technique Layer. Each layer has its unique roles, where it may crosscut between layers.

Ivkovic [6] was using Formal Concept Analysis (FCA) to find the dependencies in the intensive software system. The automation of dependency management combines with the Model Driven Architecture (MDA). In the FCA, features are grouped based on similar functionality and also correspond to the characteristic of software model. The identity of cluster objects which share common attributes as FCA objects makes it possible to have clusters of feature. This approach heavily managed by modeling the business process. However, the dependency between elements of model will bloat the feature model.

In the web services, Godse [7] proposed the ANP as tool to manage the selection of services. It supports the selection of match services as quantitative approach to correspond the interdependent relationship. Godse's approach emphasizes the multi-criteria decision making to answer the complexity of dependency between services that affect the functionality of the service request in a whole.

4. Research Method

This research is the cultivation of our previous research [8, 9,10], with extension on quality attributes that focuses on non-functional factor dependencies. Beforehand, we reviewed several approaches in Software Product Line decision modeling. And, found that the feature model must also answer the quality attributes of the requirements, comprehensibly. In the SPL development, features of feature model contain functionality that may end up, for some cases, as quality attributes [11].

In our approach, as illustrated in Figure 1, it starts by mapping the features of feature model into specific quality attributes. This step analyzes the possibility of features functionality as non-functional factors when it's grouped with specific constraints. After that, the dependency between groups and elements of groups (features) are managed in the graph as network. Subsequently, the sub network that manages the non-functional factors is also visualized in a graph. When the relationship of dependency already detailed, we analyze the dependency in global network. Based on the expert analysis, the value of each

quality attributes are mapped into the ANP approach. After that, the possibility of match dependency that may affect the product configuration is executed. In order to have a reasonable outcome, the quality attributes dependencies and features dependencies are having different analysis process. Ultimately, the whole dependency will be explicated in the multi scale empiric data.

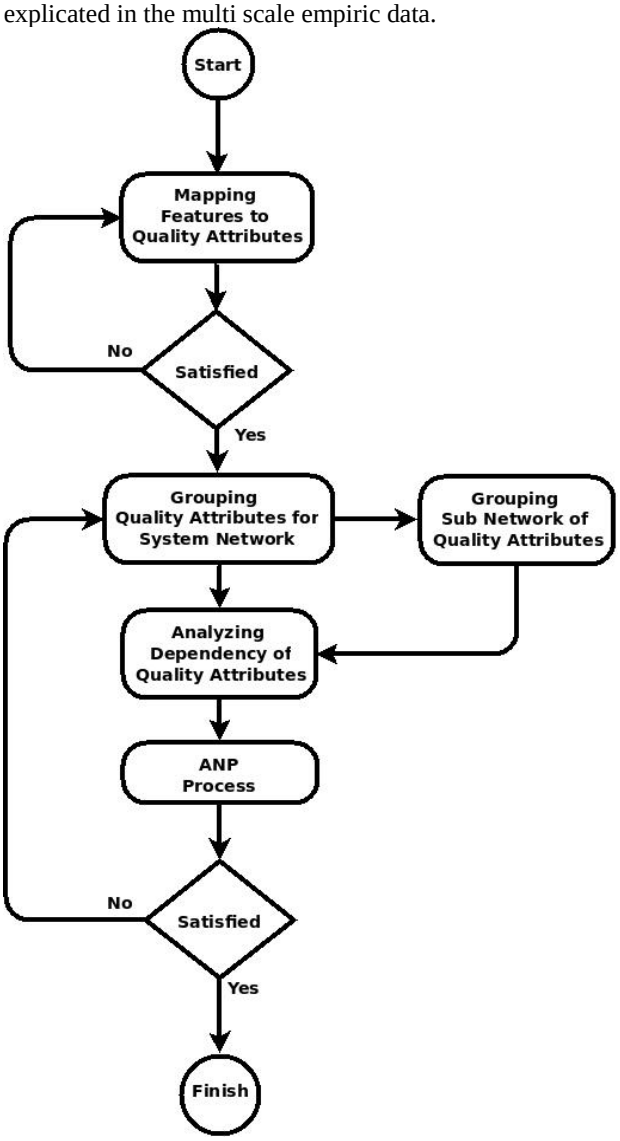


Figure 1. Hybrid-ANP approach

The dependency between quality attributes may end up as a whole as relationship graph. Where, the dependency may be represented as follows:

a. Feature to feature dependency.
b. Feature to Quality attribute dependency.
c. Quality attribute to Quality Attribute dependency.

In this approach, the judgment of multi scale data in the ANP was done by the Software Product Line expert. And, the dependency between feature and quality attribute is based on the features dependency of features of a feature model.

5. Case Study

In this case study, we implement the hybrid-ANP approach to configure the eLearning Product Line (eLPL). Where, in an eLearning System the quality attributes reveals the capability of system to support the university environment. Several quality attributes that has emerges such as, Usability, Security and Reliability. The Usability describes the easiness or easy to follow behavior of the system. This quality attributes makes the preference of design should be able to increase the user acceptance of object or topic. In the Security, it mainly focuses of how to make sure everything that concerning to privacy will be secured. The Reliability is the degree of trust that must be supported or provided by a system in design. In the eLearning system, reliability correspond to an endurance of performance that always stable during heavy load or daily basis activity, for example the evaluation and marking procedures in the eLearning must always be the same, although it is used by day or night, and do not change the learning mark of each student.

The first steps of our work begin with features grouping of eLPL feature model, then apply simply flattening process as shown in Figure 2. The eLPL feature model, the features are formed in a hierarchy structure. We could not group the features if it is still in a feature model hierarchy. To address this shortcoming, we group the features into the same level, without leaving the original structure. To this, we shrink the information of features in the feature's label.

As seen in Figure 5, the feature model of eLearning Product Line (eLPL), the full black circle indicate Mandatory – it means the feature must be included into the product configuration. And, the empty circle as Optional or Variant, that means the feature may be selected or not. In this feature model, we do not include the optional cardinality. Meanwhile, the optional branch of the feature model were flattening [12, 13] which were reduced the highest degree of tree's branch. Ultimately, the k=0 (degree of flattening) is shown in Figure 6.

The whole steps of flattening must consider the previous branch of features information. To keep this information, we use the labeling system for each flattened features. Further, features of feature model that already been flattened must always be linking to the feature property. In this paper, we do not explain about feature property in detail.

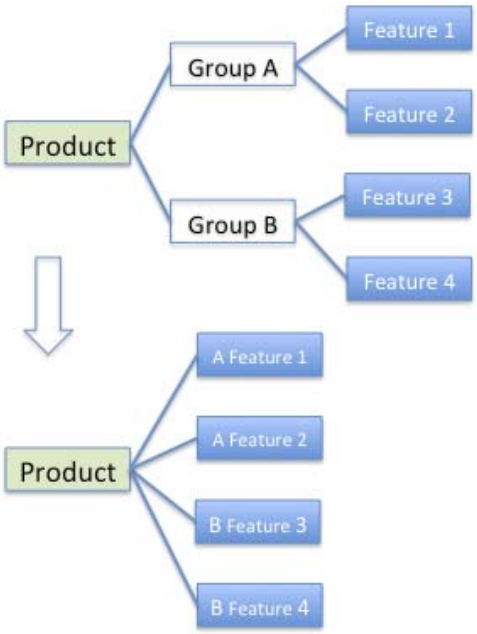


Figure 2. Illustration of Flattened Features

Based on the *k=0* flattening feature model, the groups of quality attributes are judged by expert. Subsequently, the expert of SPL generates the scale factor of Security, Usability and Reliability as shown in Table 1, Table 2 and Table 3, respectively.

Table 1. Security Scale Factor

Feature	Score	Priority
Authorization Internal	27.28%	2
Authorization External	13.22%	4
Enroll With Key	34.28%	1
Enroll With No Key	4.82%	6
Account with Single Add	14.13%	3
Account With Bulk Load	6.26%	5

Table 2. Usability Scale Factor

Feature	Score	Priority
Context Single	8.99%	3
Context Multiple	42.40%	1
Folder Internal	8.48%	4
Folder External	40.13%	2

Table 3. Reliability Scale Factor

Feature	Score	Priority
Data Management Single	17.44%	3
Data Management Share	12.50%	4
Authorization Internal	34.82%	1
Authorization External	18.35%	2
Context Single	6.46%	6
Context Multiple	10.43%	5

In the next step, the features of flattened feature model are grouped into specific quality attribute (Security, Reliability, and Usability). The result of our grouping process, as follows:

Security:

1. Account with Bulk load + Authorization Internal + Enroll With Key as local Verification (**LV**)
2. Account with Bulk load + Authorization Internal + Enroll With No Key as Open Verification (**OV**)
3. Account with Bulk load + Authorization External + Enroll With Key as Network Verification (**NV**)
4. Account with Bulk load + Authorization External + Enroll With No Key as Open Verification (**ONV**)
5. Account with Single Add + Authorization Internal + Enroll With Key as High Level Local Security (**HLLS**)
6. Account with Single Add + Authorization Internal + Enroll With No Key as Low Level Local Security (**LLLS**)
7. Account with Single Add + Authorization External + Enroll With Key as High Level Network Security (**HLNS**)
8. Account with Single Add + Authorization External + Enroll With No Key as Low Level Network Security (**LLNS**)

Usability:

1. Context Single + Folder Internal as Advance Single Context (**ASC**)
2. Context Single + Folder External as Easy Single Context (**ESC**)
3. Context Multiple + Folder Internal as Advance Multi Context (**AMC**)
4. Context Multiple + Folder External as Easy Multi Context (**EMC**)

Reliability:

1. Authorization Internal + Context Single + Data Management Single as Very High Level Security Single Context Management (**V – HLS – SCM**)
2. Authorization Internal + Context Single + Data Management Multiple as Reliable High Level Single Security Context Management (**R – HLS – SCM**)
3. Authorization Internal + Context Multiple + Data Management Single as Very High Level Security Multiple Context Management (**V – HLS – MCM**)
4. Authorization Internal + Context Multiple + Data Management Multiple as Reliable High Level Security Multiple Context Management (**R – HLS – MCM**)
5. Authorization External + Context Single + Data Management Single as High Level Security Single Context Management (**HLS – SCM**)
6. Authorization External + Context Single + Data Management Multiple as Reliable Single Context Management (**RSCM**)
7. Authorization External + Context Multiple + Data Management Single as High Level Security Multiple Context Management (**HLS – MCM**)
8. Authorization External + Context Multiple + Data

Management Multiple as Reliable Multiple Context Management (**R – MCM**)

The feature model is a hierarchy structure. To represent dependency, feature model creates crosscut relationship. However, the most difficult in representing the quality attributes of a feature model is the dependency between features. In our approach, the quality attributes are having dependencies in a network form. As shown in Figure 6, the Usability have dependency between its non-functional factor.

The quality attributes dependency's describes the sub-network of multi scale criteria. Each sub-network interacts in a global network. Where, each network will have its matrix. And, the global network will establish the global network. The trade-off of this approach is emphasized on the network evolution. Further, the network evolutions emerge as of the feature configuration select or remove the feature, during the process. To deal with the evolution, each configuration result will also include into the product configuration as well.

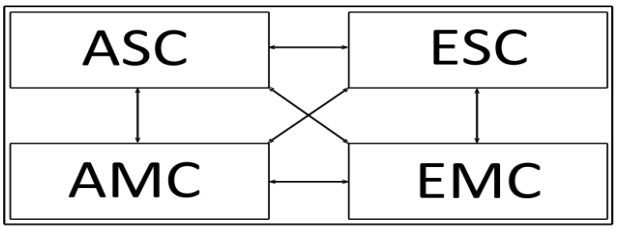


Figure 3. Usability Dependencies

The dependency between quality attributes in the specific attributes is called sub-network. The example of sub-network is shown in Figure 3, the Usability have four non-functional factors which are the “Advance Single Context” (ASC), the “Easy Single Context”(ESC), the “Advance Multi Context”(AMC) and the “Easy Multi Context” (EMC). In addition, the ASC, ESC, AMC and EMC are also grouped from features of eLearning Product Line feature model.

Each quality attributes have their matrix to describe dependency. In this step, the pair wise comparison matrix is created by taking into account the scale of value between quality attribute dependencies. Furthermore, the features that are grouped into non-functional factors are also judged to have multi scale factor. The difference between quality attribute scale factor and the group of features is that, the quality attribute scale factor includes the importance and the impact of dependency. Meanwhile, the multi scale factor groups of features only judge the scale factor without dependency. In the multi scale judgments for groups of features, the AHP method was adopted.

As illustrated in Table 4 and Table 5, respectively. The pair wise comparison matrix represents the level of importance between quality attributes in the network. It describes the dependency of quality attributes, which eventually, also describes the affect of certain quality attributes to the structure of product line configuration properties.

The dependencies between quality attributes affect the decision of product configuration. In eLearning Product Line configuration, the changes of which features will be included into a configuration will affect the quality attribute structure. And, will also affect the global dependency. The ANP focuses on the relationship between items in the network. Where, the dependency must be fulfilled by the property feature, which already grouped into the quality attributes. Mostly, the ambiguity emerges when the quality attributes do not have sufficient information. In this case we see that the knowledge management may be helpful in the next research.

Table 4. Security Pair wise Comparison Matrix

	LV	OV	NV	ONV	HLLS	LLLS	HLNS	LLNS
LV	1	6	4	8	1/2	5	2	7
OV	1/6	1	1/4	3	1/7	1/3	1/5	2
NV	1/4	4	1	6	1/5	3	1/3	5
ONV	1/8	1/3	1/6	1	1/9	1/5	1/7	1/2
HLLS	2	7	5	9	1	6	4	8
LLLS	1/5	3	1/3	5	1/6	1	1/4	2
HLNS	1/2	5	3	7	1/4	4	1	6
LLNS	1/7	1/2	1/5	2	1/8	1/2	1/6	1
sum	4.38	26.83	13.95	41	2.50	20.03	8.09	31.50

Table 5. Usability Pair wise Comparison Matrix

	ASC	ESC	AMC	EMC
ASC	1	3	4	7
ESC	1/3	1	12	4
AMC	1/4	1/2	1	3
EMC	1/7	1/4	1/3	1
sum	1.73	4.75	7.33	15

The global dependency of quality attributes is called network in the ANP approach. The dependency of this case study, as shown in Figure 4, describes the relationship between quality attributes. This dependency is called Global Network. In each element of global network consist of quality attributes that are grouped based on its goal of development.

To describe the internal relationship of quality attributes, the detail of internal quality attributes must also be taking into account. In the ANP, the internal relationships have its specific multi scale factor calculation. Where, it uses Analytical Hierarchy Process (AHP) approach. In the internal relationship, one quality attribute may influence others. In this case, we need to calculate not only the relationship in a single node of global network.

We also need to represent the decision of quality attribute in the internal relationship to another internal quality attribute element of different quality attribute elements. For example, the dependency between Security and Usability should be clear. In this approach, the relationship must have its multi scale factor, to help the product line engineer consider the last configuration will be. The internal and global dependency pair wise matrix is illustrated in Table 6, Table 7, Table 8

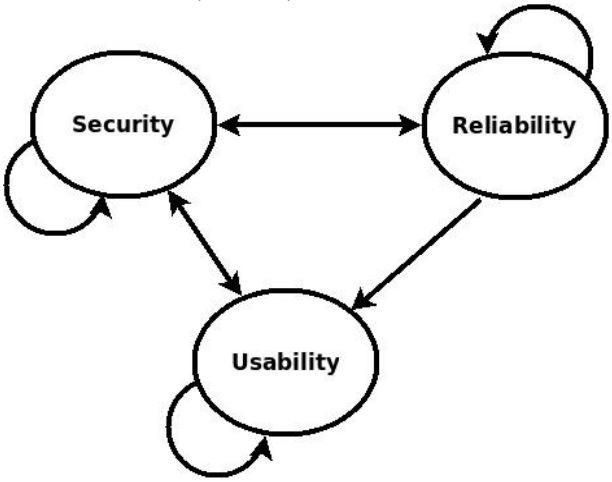


Figure 4. eLearning Product Line Network Model

Table 6. External Dependency Reliability to Security

External	HLNS	NV	ONV	LLNS	Priority
HLNS	1	3	7	6	0.55
NV	1/3	1	6	5	0.33
ONV	1/7	1/6	1	1/2	0.06
LLNS	1/6	1/5	2	1	0.09

Table 7. Internal Dependency Reliability to Security

Internal	HLLS	LV	OV	LLLS	Priority
HLLS	1	2	7	6	0.51
LV	1/2	1	6	5	0.33
NV	1/7	1/6	1	1/2	0.05
LLLS	1/6	1/5	2	1	0.10

Table 8. Dependency Reliability to Usability

Single	ASC	ESC	Priority
ASC	0.25	0.25	0.25
ESC	0.75	0.75	0.75

The Internal and external dependency as shown in Table 6 and Table 7, represent the support of information to the Software Product Line engineer that the dependency of quality attributes between reliability and security have impact to both quality attributes. In this case, the global dependency can be represented in a table. The dependency of global network is shown in Table 9.

Table 9. Super Matrix eLearning Product Line Quality Attribute

Initial Super Matrix		Goal		
		Security	Usability	Reliability
Security	**LV**	0.23	0.03	0.05
	OV	0.04	0.16	0.23
	NV	0.11	0.07	0.02
	ONV	0.02	0.33	0.11
	HLLS	0.34	0.02	0.07
	LLLS	0.07	0.11	0.33
	HLNS	0.16	0.05	0.03
	LLNS	0.03	0.23	0.16
Usability	**ASC**	0.56	0.06	0.16
	ESC	0.24	0.16	0.05
	AMC	0.15	0.23	0.54
	EMC	0.06	0.55	0.25
Reliability	**HLSMCM**	0.05	0.16	0.07
	HLLSCM	0.07	0.11	0.05
	RHLSMCM	0.11	0.07	0.16
	RHLSSCM	0.16	0.05	0.11
	RMCM	0.02	0.33	0.03
	RSCM	0.03	0.23	0.02
	VHLSMCM	0.23	0.03	0.33
	VHLSSCM	0.33	0.02	0.23

The trade-off dependency in the internal network affects the global network. Further, the decision of quality attributes will affect the whole configuration. In this case study, we found that the changes of selection for quality attribute's internal network will affect the whole configuration. In contrast, when there is no relationship between quality attributes, the internal network may not impact the global network. In our perception, the representation of quality attribute dependency in the network makes it easier for software product line engineer to select the suitable features for their product goal.

In the super matrix, the quality attributes are items that influence their relationship. The relationship reveals the possibility of many specific products. Further, the quality attribute dependency may impact to the final product configuration. To this, sub-network quality attributes that are formed from features of a feature model will resolve the detail of feature dependency.

The super matrix as shown in Table 9, presents the impact of a decision and its scale of influence to the whole quality attributes in eLearning Product Line quality attributes dependency. In Table 9, the rows of quality attributes (Security, Usability, and Reliability) represent the scale factor on each sub-network. And, the columns of quality attributes represent the global network multi scale, which are also the dependency between sub-networks.

6. Conclusion

We already proposed a hybrid approach to support the software product line designer to configure specific product based on quality attributes criteria. The quality attributes are judged based on its importance of non-functionality. And, the non-functionality is group of features, which have its functionality. The quality attributes dependencies reveals the group of feature dependency. In our approach, the quality attributes dependencies are considered as networks that have sub-networks. Each sub-network dependency represents groups of feature dependency, where, each dependency may affect the specific sub-network or the global network. Using the Analytical Network Process (ANP), we already proposed of how multi scale factor on each sub-network represented. Ultimately, it will help the software product line designer to configure the specific product from the member product of a product line.

Acknowledgment

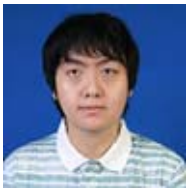
This research was funded by the Lembaga Penelitian dan Pengabdian Masyarakat – Universitas Pelita Harapan (LPPM-UPH) grant project number P-001-FIK/II/2012

References

[1] P. Clements, L. Northrop, Software Product Lines: Practices and Patterns, Addison Wesley, 2001.
[2] K. Pohl, G. Bockle, F. van der Linden, “Software Product Line Engineering: Foundation, Principles and Techniques”, Springer-Verlag, Berlin, 2005.
[3] K. Czarnecki, Generative Programming: Principles and Techniques of Software Engineering Based on Automated Configuration and Fragment-Based Component Model, PhD Thesis, Technical University Illmenau, Germany, 1998.
[4] T. L. Saaty, “Decision making with the analytic hierarchy process,” International Journal of Services Sciences, vol. 1, no. 1, p. 83, 2008.
[5] K. C. Kang, J. Lee, P. Donohoe, ”Feature-Oriented Product Line Engineering”, IEEE Software, Vol. 19, No. 4, pp.58-65, July 2002.
[6] I. Ivkovic and K. Kontogiannis, “Using formal concept analysis to establish model dependencies,” International Conference on Information Technology: Coding and Computing (ITCC’05) - Volume II, pp. 365-372 Vol. 2, 2005.
[7] M. Godse, R. Sonar, and S. Mulik, “Web Service Selection Based on Analytical Network Process Approach,” 2008 IEEE Asia-Pacific Services Computing Conference, pp. 1103-1108, Dec. 2008.
[8] I Made Murwantara, “Towards Quality Attributes Decision Modeling Approach for a Product Line Architecture”, International Journal of Computer Science and Network Security (IJSCNS), Vol. 11, No. 11, November 2011.
[9] I Made Murwantara, “Initiating Layers Architecture Design for Software Product Line”, Proceeding of Uncertainty and Reasoning of Knowledge Engineering (URKE), 2011.
[10] I Made Murwantara, “Another Architecture Style for a Product Line Architecture,” Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services - iiWAS ’11, p. 443, 2011.
[11] T. M. Dao, H. Lee, and K. C. Kang, “Problem Frames-Based Approach to Achieving Quality Attributes in Software Product Line Engineering,” 15th International Software Product Line Conference, pp. 175-180, Aug. 2011.
[12] Xiao-Lin Wang; Bao-Liang Lu; , "Flatten hierarchies for large-scale hierarchical text categorization," Fifth International Conference on Digital Information Management (ICDIM), pp.139-144, 5-8 July 2010, DOI: 10.1109/ICDIM.2010.5664247
[13] Malik, H.H., "Improving Hierarchical SVMs by Hierarchy Flattening and Lazy Classication", In Large-Scale Hierarchical Classification Workshop (ECIR 2010), Milton Keynes, UK, 2010.

Pujianto Yugopuspito received Engineer degree in Mechanical Engineering in 1991 from Unversitas Gadjah Mada, Indonesia; Master degree in Software Techniques for Computer Aided Engineering in 1996 from Cranfield University, United Kingdom; and Doctor of Engineering in Computer Science and Communication Engineering, specialization in Software Methodology in 2001 from Kyushu University, Japan. Since 2004 he is with Universitas Pelita Harapan. His research interest includes software engineering, formal methods, high performance computing, and mobile applications. He is a member of IEEE.

I Made Murwantara received the M.S. degrees in Computer Science from the University of Indonesia in 2002. He has been a lecturer at Informatics Department, Faculty of Computer Science, Universitas Pelita Harapan since 2004. He is, now, with Research and Development Computer Lab., Informatics Dept, Universitas Pelita Harapan, Indonesia. His research interest includes Reusability, Software Product Line Engineering and Formal Method.



Adrian Hartanto Sutomo received the Bachelor of Informatics degree from Universitas Pelita Harapan in 2012.

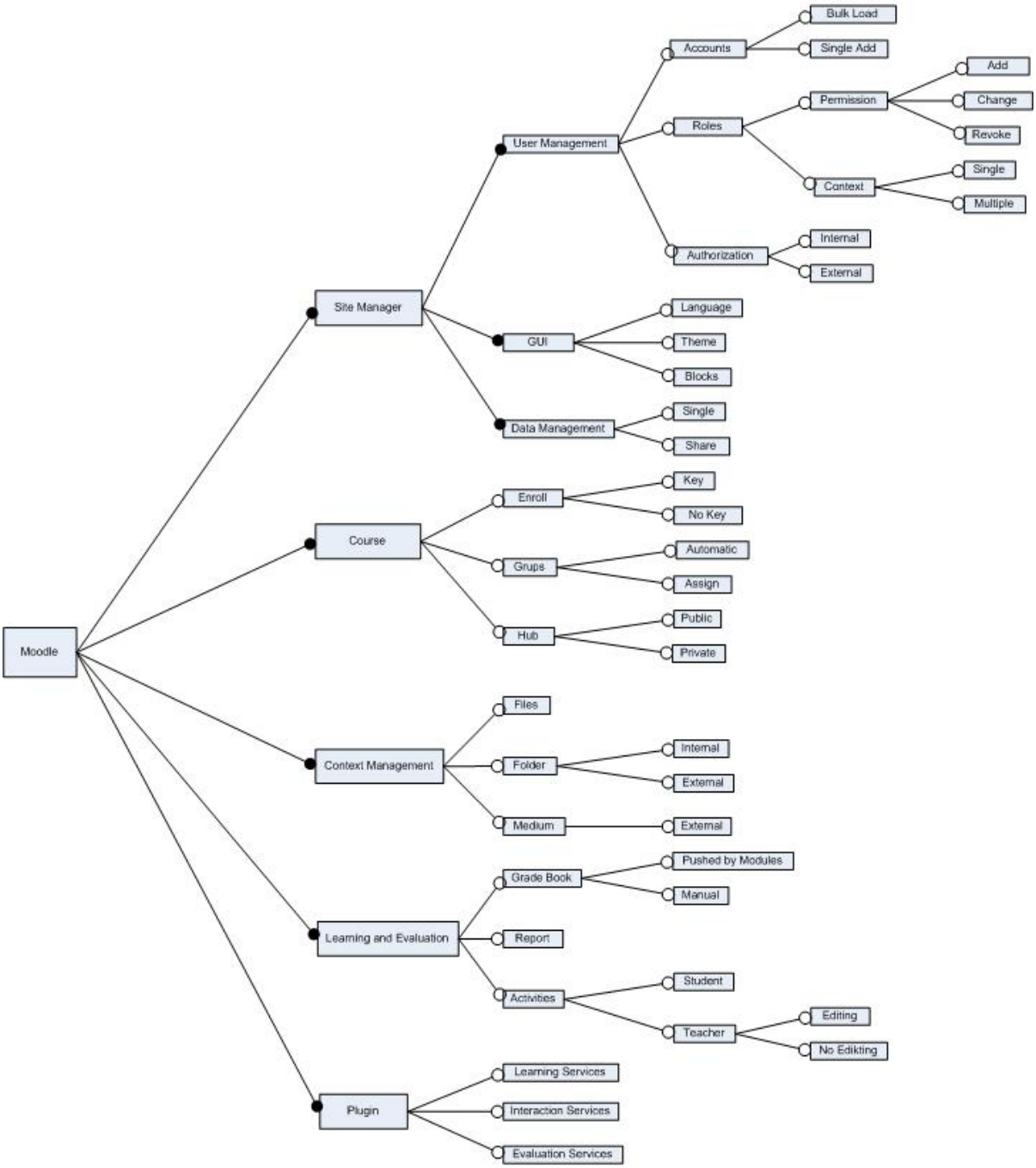


Figure 5. Feature Model of eLearning Product Line

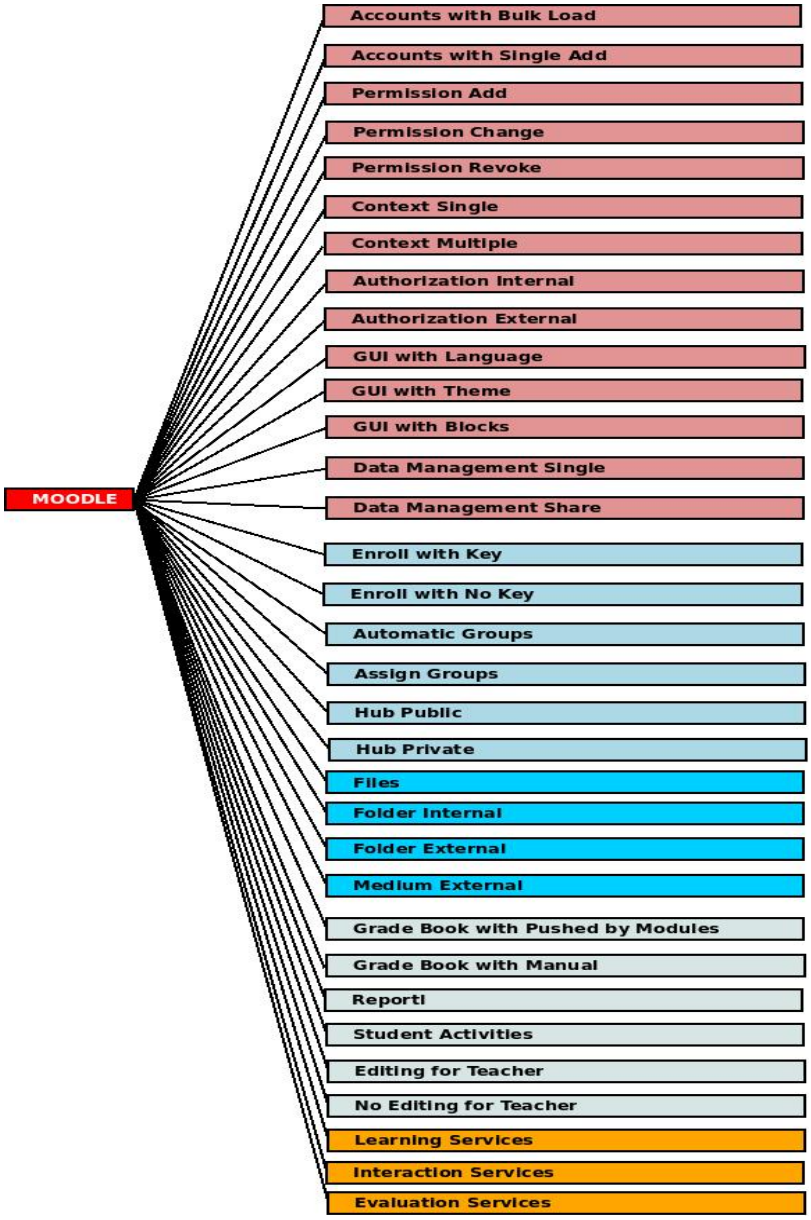


Figure 6. *k*=0 Flattening of eLearning Product Line