International Conference on Information and Convergence Technology for Smart Society
Jan. 19-21, 2016 in Ho Chi Minh, Vietnam

Optimized feature selection with crosscutting concerns in software product line feature model

Asad Abbas, Zhiqiang Wu, Isma Farah Siddiqui, Scott Uk-Jin Lee
Department of Computer Science and Engineering, Hanyang University, South Korea
asadabbas,hhhwwwuuu,isma2012,scottlee @hanyang.ac.kr

Abstract—**Software Product Line Engineering (SPLE) is best paradigm to build new products with high reusability from existing resources. Feature model is commonly used for resource management with variable and common features under product line. Key idea of feature model is to manage commonalities and variabilities of resources with mandatory, alternative and optional features. Crosscutting concerns are tangled and scattered in overall system which reduce the reusability of resources because of dependency with other features. At modeling level, the identification of these features is an important task to construct them as a separate aspect module. As a result, the addition or deletion of these aspects do not affect the system. In this paper we present: (1) an approach to find crosscutting concerns by using union-find algorithm, (2) genetic algorithm for optimized feature selection with resource constraints and crosscutting concerns.**

Keywords-Software product line; Feature model; Union-find algorithm; Genetic algorithm; Optimization.

I. INTRODUCTION

Software Product Line Engineering (SPLE) is an effective approach for development of family of software. Main purpose of SPLE is reusability of common resources to achieve high productivity with less cost and time. Products are differentiated by variable feature whereas common features are important for each product in SPLE. To achieve high reusability of resources with less cost and time, commonalities and variabilities must be analyzed systematically [1].

Feature model is same as tree of all possible resources. Every feature node has a parent except root node. Feature-oriented methodology becomes effective for SPLE to manage the commonalities and variabilities. Feature model is tree structure, contains the mandatory, alternative and optional features [2]. Common features are easy to handle because of reusability, however difficult to manage the variable features. Hard binding among features makes difficult to reuse, add or delete from different products [3].

Crosscutting concerns are scattered and tangled in multiple features which makes hard to decouple and reuse in all products under SPLE scope. Crosscutting concerns are those on which more than one features are dependent, spread on whole system and tangle with other features. Modularization technique is used for crosscutting concerns to remove the scattering and tangling of code in system. In feature model of SPL, crosscutting concerns must be identified for high reusability of features [1]. Crosscutting concerns must be identified at modeling level and get modularize at coding level. In complex feature model where exist large number of features and complex relation among them, it is difficult to identify crosscutting concerns. Identification of crosscutting concerns in feature model is still an ongoing problem.

In this paper we have used Union-find algorithm to identify the crosscutting concerns in feature model. The contribution of this paper is to search crosscutting concerns by using Union-find algorithm and optimize the feature selection using genetic algorithm.

- We show union-find algorithm to find crosscutting concerns in SPL feature model.
- We describe every possible case of relationship among features in feature model which cause the crosscutting.
- Cost and efficiency may get effected because of existing crosscutting concerns and from how many features they are called. We optimize the possible solution under given constraint by considering the crosscutting concerns by using genetic algorithm.

Our work presents the large scale feature model with complex relationship of features and shows how to mining crosscutting concerns and get the optimum solution based on crosscutting concerns.

The rest of paper is organized as follows: Section II contains background, section III contains brief introduction on feature model, section IV contains importance to identify crosscutting concerns, section V gives brief explanation on union-find algorithm, section VI describes genetic algorithm and section VII presents conclusion.

II. BACKGROUND

There are several techniques available to find the crosscutting concerns.

Jianmei Gue et. al discussed genetic algorithm for optimize feature selection under constraints of resources. Authors discussed simple case study of a database and optimized the best solution for constraints of cost, memory and CPU of resources [4].

Mohammad Mhawish et. al discussed genetic algorithm to identify the crosscutting concerns. They proposed a method by using static analysis and vector-space model [5].

Şerban, Gabriela et. al discussed Graph algorithm to identify the crosscutting concerns. They introduced Graph Algorithm in Aspect Mining (GAAM) [6].

Şerban, Gabriela et. al discussed k-means clustering algorithm to identify crosscutting concerns by identifying the similarities of objects [7].

In our work, we have applied union-find algorithm to identify the crosscutting concerns in feature model of SPL. Feature model contains feature constraints with optional, alternative and mandatory features. We have discussed the genetic algorithm for optimization of feature selection with crosscutting concerns in feature model.

III. Feature Model

Simple automobile feature model with small number of crosscutting concerns is defined as case study to find the optimum values and find the effectiveness and efficiency of UACF. In this case study, XLT (eXtra Large Trim), speed sensor and crash sensor are the crosscutting concerns. These crosscutting concerns may cause tangling and scattering in large scale automobile feature model. To optimize such crosscutting concerns in large scale feature model is a difficult task. In literature, many algorithms are discussed to optimize the crosscutting concerns "Fig. 1".

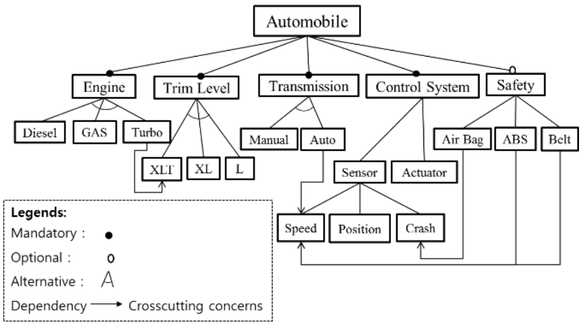

Fig.1 Feature Model of "Automobile"

IV. Identify Crosscutting Concerns

Identifying a perfect crosscutting concern in feature model is still an ongoing problem. Identification of correct crosscutting features in simple feature model is a simple task. However in large and complex feature model where large number of features and relationship among them is also very complex, makes hard to identify the optimal values related to crosscutting concerns [6].

Crosscutting concerns are scattered over system because of dependency with other features [6]. In feature model to find the crosscutting concerns, Union find optimization algorithm is more efficient and easy to use.

V. Union-find Algorithm

In an Union-find algorithm, features are connected with disjoint-set data structure also called as union-find data structure. Disjoint-set data structure keeps the track of elements in different portioned data set into number of disjoints. Each element in data structure has a pointer which indicates his own disjoint-set and one pointer for itself. Elements from same disjoint-set can be traced by root pointer which is identified as a root element. Union operation is used to combine two disjoint-set by pointing of root element of one disjoint-set to an element of other set [1].

Union-find algorithm is effective and simple to find crosscutting concerns in feature model because of disjoint-set data structure or merge the elements from different sets.

Union Find Algorithm for searching in Feature Model

Merge(a,b) First step

1. Input: the number of features and feature model with F={fi} 1 ≤ i ≤ n RF (Relationship of features)
2. S' = x U y
3. x.total <-- get the children number both x and y
4. y.parent <-- setting the parent of y is x
5. x.children <-- setting the children of x is y
6. Output: Union of all disjoint sets

Find_Crosscutting(n) Second Step

```
foreach feature ∈ N do
    numParent <-- amount of parents
    numChildren <-- amount of children
    If numParent>1 and numChildren>1 then
notTerminalCrosscutting(feature);
    else
        if numParent>1 and numChild =0 then flag<-
    true
        else  JugeParentIsCC(feature);
        end
    end
end
Output: crosscutting features
```

If this feature is not terminal feature, it has more than 1 parents or children jump this.

VI. GENETIC ALGORITHM OPTIMIZATION

GA is random search that imitate evaluation, optimize best solutions in vast population. High constrained problems are solved very well by using GA. Random selection of chromosomes find the fitness of possible solution [4, 5].

Optimization of best solutions under cost and feature constraints is a difficult task where Fan-In of crosscutting concerns is very high. High Fan-In crosscutting concerns calls from many methods which effects on wrong optimize solutions. "Fig. 1" ABS (Auto Break System) feature has dependency on "Sped" and "Position", these features may ignored during optimization under constraints of cost<500. Identification of crosscutting concerns is important before optimization to get best optimum solution.

To overcome challenges of optimization, GA is best approach to find optimum solutions with lots of crosscutting concerns in feature model. In this section we detail discuss GA specifications and how to adopt for optimization feature selection with resource constraints and crosscutting concerns.

Genetic Algorithm for optimized feature selection

1. Generate feature model with n resources F
   F = {fi}, 1 ≤ i ≤ n, C (Constraints), R (Feature), V(F) value of features, F(R) feature relationship and Rc (resource constraints).
2. Evaluate the fitness f(i) of each feature i in the feature model.
3. Set of new features according to fitness of each.
4. Select two parent p1 and p2
5. Crossover p1 and p2 to get offspring
6. Mutation to get new offspring (update set of features).
7. Population of new offspring
8. Use new generated population for a further run of the algorithm
9. If required optimized feature selection satisfied then stop, otherwise repeat from step 2.

In a small feature model with 200 total number of resources which consist on mandatory, alternative and optional features. Mandatory are important features cannot be ignored because of compulsory part of every product, no need to optimize these features. Optimization is required for optional and alternative features. Crosscutting concerns also important part of feature model because so many other features dependent on these features.

F=f(i) here i=200, MF (mandatory features)=50, CC (Crosscutting Concerns)=50, sum of Alternative and optional features=100

In this case we set the cost for all crosscutting concerns i.e. average cost is “x” for each crosscutting concerns. In result optimized feature selection is required on alternative and optional feature i.e. 100 rather than whole feature model. By this process we achieve best feature selection with less searching in large scale feature model where the number of resource constraints are more.

VII. Conclusion

In this paper we have presented union-find algorithm for searching crosscutting concerns from feature model and apply genetic algorithm for optimized feature selection with resource constraints and crosscutting concerns.

Searching of crosscutting concerns at very initial stage of modeling is good approach. It minimizes the process and time of optimization of feature selection and increase the efficiency and accuracy with given resource constraints of product. We have applied union-find algorithm and genetic algorithm of small feature model with resource constraints cost ≤ 500. We set cost of crosscutting concerns which we found by union-find algorithm cc ≤ 20.

In future, we will apply our approach on 10 to 10000 resources of feature model and do the comparative study on evaluation results.

References

[1] Lee, Kwanwoo, Kyo C. Kang, Minseong Kim, and Sooyong Park. "Combining feature-oriented analysis and aspect-oriented programming for product line asset development." In Software Product Line Conference, 2006 10th International, pp.10-pp. IEEE, 2006.

[2] Thüm, Thomas, Christian Kästner, Fabian Benduhn, Jens Meinicke, Gunter Saake, and Thomas Leich. "Featureide: An extensible framework for feature-oriented software development." Science of Computer Programming 79 pp.70-85, 2014.

[3] Tizzei, Leonardo P., Cecilia MF Rubira, and Jaejoon Lee. "An aspect-based feature model for architecting component product lines." Software Engineering and Advanced Applications (SEAA), 2012 38th EUROMICRO Conference on. IEEE, 2012.

[4] J Guo, J White, G Wang, J Li, Y Wang "A genetic algorithm for optimized feature selection with resource constraints in software product lines." Journal of Systems and Software 84.12 pp.2208-2221, 2011

[5] Mohammad Mhawish, Manjari Gupta. “A New Genetic Algorithm Tool for Clustering-Based Aspect Mining Using Static Analysis and Vector-Space Model” International Journal of Scientific Research Engineering & Technology (IJSRET), vol4, Issue4, pp.434-440, 2015

[6] Serban, Gabriela, and GRIGORETA SOFIA Moldovan. "A Graph Algorithm for Identification of Crosscutting concerns." Studia Universitatis Babes-Bolyai, Informatica, vol.LI (2) pp.53-60, 2006

[7] Şerban, Gabriela, and Grigoreta Sofia Moldovan. "A new k-means based clustering algorithm in aspect mining." Symbolic and Numeric Algorithms for Scientific Computing. SYNASC'06. Eighth International Symposium on. IEEE, pp.69-74, 2006