Article

Graphical matching rules for cardinality based service feature diagrams

Faiza Kanwal[1], **Ghulam Mustafa Assad**[1], **Arif Iqbal Umer**[1], **Muhammad Naeem**[2], **Sheraz Ahmad**[1]
[1]Department of Information Technology, Hazara University, Mansehra, Pakistan
[2]Department of Information Technology, Abbottabad University of Science and Technology, Abbottabad, Pakistan
E-mail: ms_cs41@yahoo.com, gm_assad@yahoo.com, arifiqbalumar@yahoo.com, naeem@aust.edu.pk, sheraz.cse15669@yahoo.com

Received 26 April 2016; *Accepted 31 May 2016*; *Published online 1 March 2017*

Abstract
To provide efficient services to end-users, variability and commonality among the features of the product line is a challenge for industrialist and researchers. Feature modeling provides great services to deal with variability and commonality among the features of product line. Cardinality based service feature diagrams changed the basic framework of service feature diagrams by putting constraints to them, which make service specifications more flexible, but apart from their variation in selection third party services may have to be customizable. Although to control variability, cardinality based service feature diagrams provide high level visual notations. For specifying variability, the use of cardinality based service feature diagrams raises the problem of matching a required feature diagram against the set of provided diagrams.

Keywords feature modeling; cardinality-based service feature modeling; service feature diagrams; matching; software product line.

Selforganizology
ISSN 2410-0080
URL: http://www.iaees.org/publications/journals/selforganizology/online-version.asp
RSS: http://www.iaees.org/publications/journals/selforganizology/rss.xml
E-mail: selforganizology@iaees.org
Editor-in-Chief: WenJun Zhang
Publisher: International Academy of Ecology and Environmental Sciences

1 Introduction

Product line engineering is one of the ways which researchers used in industry to develop product of the product line. The problem researchers faces for artifact development, is the use of commonality and variability among the features of any products. Feature modeling is traditional notation to deal with such type of problem (Batory, 2005). The visualization of Service-oriented Architectures is to support supple on-demand connections in a souk of request components where requestors' necessities are coordinated matched aligned with providers' tender. Computerization of for capful invention, choice and necessary of services prolong to cause key confront. Among these challenges confront one is level of flexibility required to locate a service to exact necessities For instance, a requestor while scheduling to organize a party. We may not only be interested in booking a hall in a hotel but also to decorate hall, arrangement of transport and lunch. No provider can provide all of these requirements. In such situation when one provider cannot provide all these requirements, then two conditions are possible. Hire more than one provider or leave any one of requirement. The major aim of this paper is to address the matching of cardinality based feature diagram (CSFD) to confine mutually the requestor's and provider's flexibility individual, along with extra characteristic, needed and voluntary requirements. The analysis leads us to the conclusion that CSFD should be interpreted as linear logic formulas, and that deduction in linear logic provides the semantics for matching. The problem of matching one goal against several offers is addressed both statically, where all offers are known in advance, and dynamically, where one offer is chosen and matched at a time and a remainder goal is constructed for matching with future offers. The rest of the paper is arranged as follows section 2 contain the background of feature modeling, section 3 contain the matching challenges of cardinality based service feature diagram.

2 Background

To sustain reusability of feature (Kang et al., 1990), introduce feature diagrams known as (FODA) feature oriented domain analysis. Feature oriented domain analysis use three different types of feature diagram, 1) mandatory feature must be selected when the parent is selected; 2) Optional feature may be selected, or 3) discarded when the parent feature is selected. In alternative sub group exactly one sub feature must be selected when its parent is selected. Feature model represent, feature of a product in a tree like structure to characterize variability and commonality, among feature of software product line. Here we use them to describe variability of services while considering the notations of (Czarnecki and Eisenecker, 2000). In product line all type of feature grouping has been expressed by a single feature diagram. The selection of feature under some given permission is called an instance.

Different author proposed many different extensions of original feature diagram. For example Feature-RSEB proposed in (Czarnecki et al., 2005), is the first extension of original feature diagram. Cardinality based feature diagram proposed in (Czarnecki and Wasowski, 2007), is the 2nd extensions of feature diagram. Next extension of FODA diagrams is (SFD) stand for service feature diagram introduced in (Naeem and Heckel, 2009). Cardinality based service feature diagrams, introduced in Assad et al. (2015), is the latest extension of FODA diagram (Fig. 1).

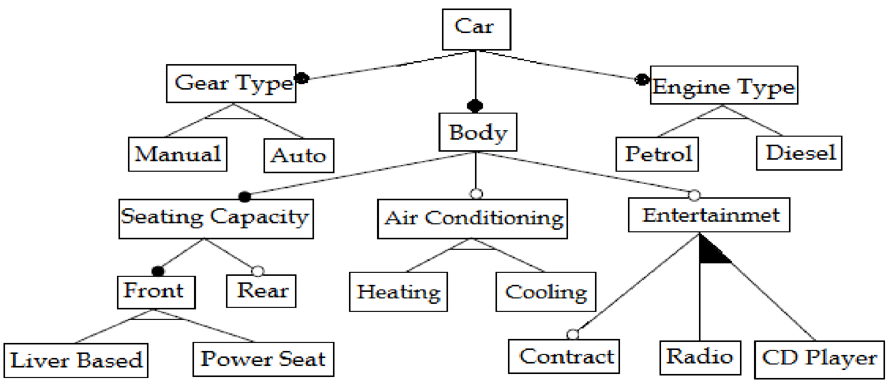

Fig. 1 A sample feature model.

1. Child feature is selected only in case, when its parent must also be selected.
2. Mandatory features must be selected whenever its parent is selected.

3. Optional feature may select or reject when its parent is selected.
4. Exactly one sub feature must be selected of its alternative group, when its parent is selected.
5. If a parent is chosen, at least one sub feature of it's Or group must be selected.

3 Matching Challenges

To confirm that a collection of services offered by different provider together, satisfy the need of a requestor, we have to match their specifications (Heckel and Cherchago, 2007). Specifying objective, suggest by means of feature models and take description of the variability allowable for and required by requestors and provider. From the matching of feature diagrams, we mean to state formally when the requirements and description are modeled by service feature diagrams

3.1 Matching of models

In this section, we will discuss the matching of general models. Traditional approaches for model matching consider them as trees and use dedicated algorithms to get the difference between those models. This difference is computed by matching the elements of one model with the elements of other. The algorithms used for the matching of models reside in one of the following categories: Static-identity (Gunter and Kuhn, 1999), in this category the basic approach is to identify the matched elements of models based on id and it is assumed that each one model element has a fixed static identity. Signature-based (Kang et al., 1990): the matching of elements is done on the basis of their signatures, which are calculated dynamically by a model query language using the values of its features. Similarity-based (Kang et al., 1998): this type of treat models as typewrote characteristic graphs and matching of basics is founded on the combined comparison of the features this type affects matching algorithms based on an exacting modeling language. Lee et al. (2008) argued that none of the above categories provide satisfactory solution to model matching but that the trouble should be treated by make a decision on the best trade-off within the limitation imposed in the context, and for the exacting assignment at chance. There are some efforts in the literature for the composition merging of models. Lee et al. (2008) used the Epsilon Merging Language to automatically compose a model from two separate models, whereas Erl et al. (2009) provided a survey on software merging.

3.2 Matching of feature diagrams

Naeem et al. (2010) provided an algorithm to get the semantic difference of two feature diagrams. For example, if SP and SR are the semantics of provider and requestor feature diagrams, their algorithm gives $D = SP - SR$, the clauses which exist in SP but not in SR. Indeed this difference, if non-empty, can be used as remaining requirement to match with the next available offer. They use the Propositional Logic-based semantics of feature diagrams, where SP and SR are considered to be in conjunctive normal form (CNF). Narendra et al. (2008) have proposed to use feature models for the automated composition of components to capture the variability of component interfaces. They allow variability at provides interface of a component, whereas a requires interface does not have any variability. To check whether the descriptions at the provides interface satisfy the requirements at a requires interface, they check the inclusion of FRS in FMI, where FRS (Feature Requirements Specification) represents the clients requirements and FMI (Feature Model Instance) is the set of all the instances of the provider feature diagram. If FRS is contained in FMI, they proceed to bind the components. For the implementation of their approach they used a Java-based environment called Columbus.

3.3 Matching rules for service feature diagram

Graphical rules provide a simple way to present the required semantics while hiding the logical semantics. So, we define a set of graphical rules to check whether a set of descriptions satisfies given requirements. For example, if Veg and N-Veg are the features representing vegetarian and non-vegetarian food respectively we have the Fig. 2

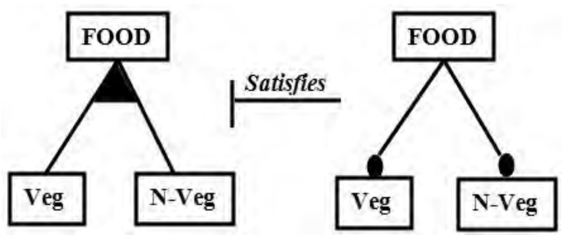

Fig. 2 Veg and N-Veg are the features representing vegetarian and non-vegetarian food.

Fig. 2 means that a requirement of an And-group of vegetable and non-vegetarian food can be satisfied by an Or-group (R) of vegetarian and non-vegetarian food.

In general, for the graphical rules, the total number of features on the left and right side of each rule should be the same. Since no matching rule deletes a feature, the number of features in description and requirements remain constant. The graphical rules can be used if multiple offers contribute towards the requestors needs. A generic rule consists of a generic feature diagram capable of capturing multiple concrete rules.

4 Cardinality Based Service Feature Diagram

As cardinality based service feature diagram the extended form of SFD, was introduced in Rao et al. (2006). There is no existing solution for their matching and merging. The main objective of this research paper is to address the above gap through proposing graphical matching and merging rule for cardinality based service feature diagram.

Our approach is more purposely targeted at the matching of service providers' descriptions against requestor's requirements. Matching numerous descriptions, variability may occur from how they are combined rather than the individual services themselves.

4.1 Graphical matching rule for cardinality based service feature diagram

Graphical rules provide a simple way to present the required semantics while hiding the logical semantics. So, we define a set of graphical rules to check whether a set of descriptions satisfies given requirements. Each graphical rule is of the form $PFD \overset{Satisfies}{\longmapsto} RFD$ where PFD is the provider and RFD the requestor service feature diagram. In general, for the graphical rules, the total number of features on the left and right side of each rule should be the same. Since no matching rule deletes a feature, the number of features in description and requirements remain constant. The graphical rules can be used if multiple offers contribute towards the requestors needs. We provide the graphical rules for the entire feature types of cardinality based service feature diagram with the use of generic rule wherever possible. A generic rule consists of a generic feature diagram capable of capturing multiple concrete rules.

4.2 Matching rules for and-groups of solitary features

An And-group can either consist of solitary, or group features, or both.

4.2.1 Rules for And-groups of Mandatory Features

Mandatory features are not flexible in nature it must be selected whenever the parent is selected so there is no distinction of who makes the selection. Let us provide the graphical rules for requirements to be satisfied by And-groups of mandatory features.

And-group of Mandatory Features to And-group of Optional (P) Feature

An And-group of mandatory features satisfies an And-group of optional features with provider's choice where (P) means that the optional features are selected by the provider. That means, if the requestor allows the provider to choose the optional features in the requirements, the And-group of mandatory features in the descriptions satisfies this requirement. Shown in Fig. 3

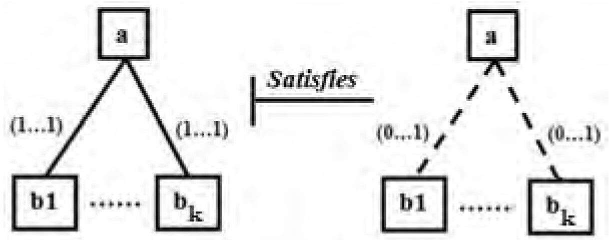

Fig. 3 And-group of mandatory feature satisfying and-group of optional (P) feature.

And-group of Mandatory Features to Group feature (P) except when condition is m=n=1: An And-group of mandatory features satisfies Group feature with provider’s choice, but this rule does not apply where m=n=1 as shown in Fig. 4.

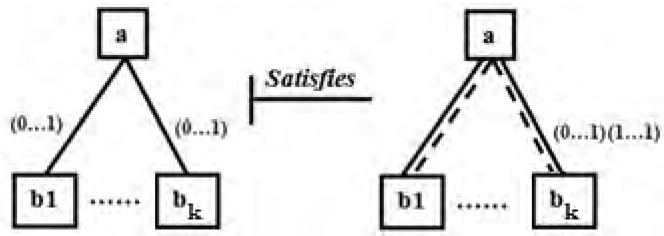

Fig. 4 And group of Mandatory Feature satisfying Group Feature (P).

4.2.2 Rules for and-groups of optional (R) features

If the parent of an optional (R) feature is selected, the requestor may select or reject an optional sub feature. In the following, we will show the cases where an And-group of optional (R) features satisfies the requirements.

And-group of Optional Features to And-group of OptP-Man Features

If a provider allows a requestor to choose from an And-group of optional features, this satisfies an And-group of OptP-man features. To know how much of the cases are captured by a generic rule, we need to replace the generic feature diagram of the rule by its corresponding concrete diagrams. For example, a generic feature diagram (OptP-Man) is indicated in Fig. 5.

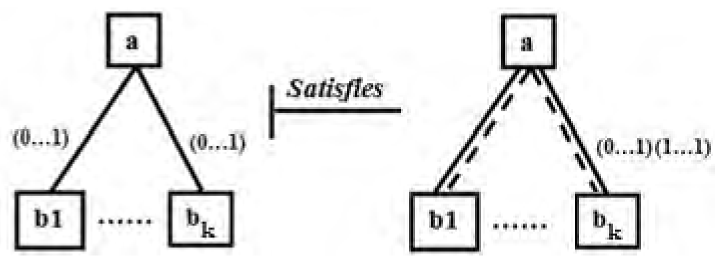

Fig. 5 And-group of Optional (R) Features Satisfying And-group of OpP-Man Features.

The used in the right of the Fig. 5 consists of two concrete rules: optional (P) feature and mandatory feature. Hence, the multi-rule shown in Fig. 5 captures the following cases:

1. And-group of optional (R) features satisfying an And-group of optional (P) features,
2. And-group of optional (R) features satisfying an And-group of mandatory features

We will use the same technique to clarify the cases captured by a multi-rule.

And-group of Optional Features to Group feature (RP)

An And-group of optional (R) features satisfies the Group feature (RP) (Group feature with requestor's or provider's choice), as shown in a generic rule of Fig. 6.

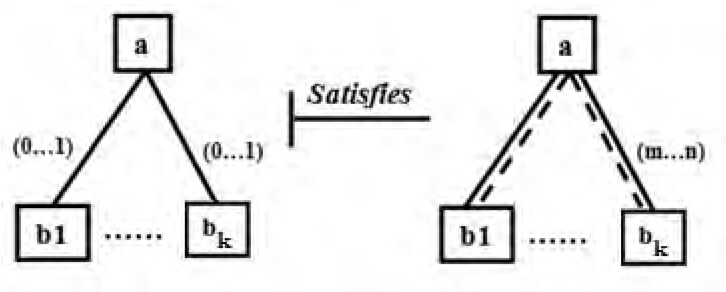

Fig. 6 And-groups of optional (R) features satisfying group feature (RP).

4.3 Matching Rules for Group feature

If the feature A is selected then feature Bm to Bk must be selected from this group in an instance, where $1 \leq m \leq n \leq k$. The selection of feature should be decided on the basis of requestor and provider preferences. So we consider two cases:

1. When a requestor makes a selection from the Group Feature
2. When a provider makes a selection from the Group Feature

4.4 Rules for requestor-based choice

Group feature with requestor's choice satisfies the following cases.

4.4.1 Group feature to and-group of optional (P) features

The Group feature (R) satisfies an And-group of optional (P) features, as shown in Fig. 7.

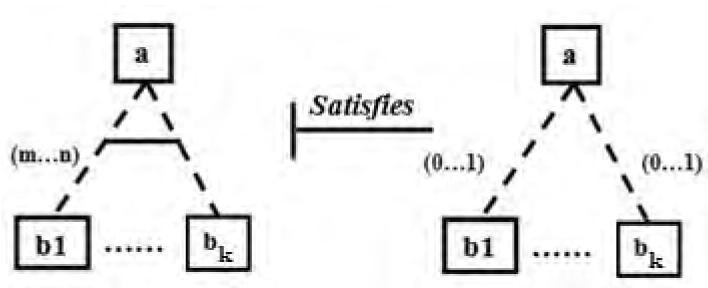

Fig. 7 Group feature (R) satisfying and-group of optional (P) features.

Group feature (R) to Group feature (P): Group feature(R) satisfies a Group feature(P), as shown in Fig. 8.

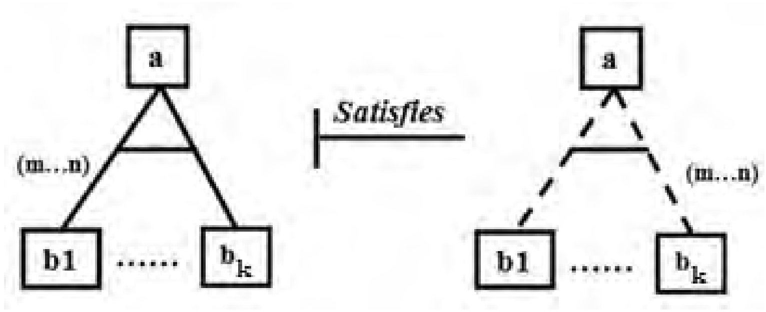

Fig. 8 Group feature (R) Satisfying group feature (P).

Group feature to And-groups: Group feature (R) satisfies the requirements of an And-group of optional (P) and Group feature (P) features respectively, as shown in Fig. 9.

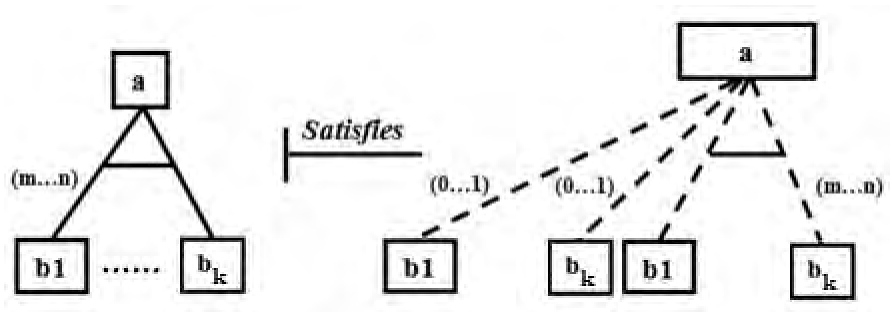

Fig. 9 Group feature (R) satisfying and-groups.

4.4.2 Rules for provider-based choice

When the provider wants to keep the choice of Group feature, it satisfies the following cases. Group Feature to And-group of Optional (P) Features: An Group feature (P) satisfies an And-group of optional (P) features, as shown in Fig. 10.

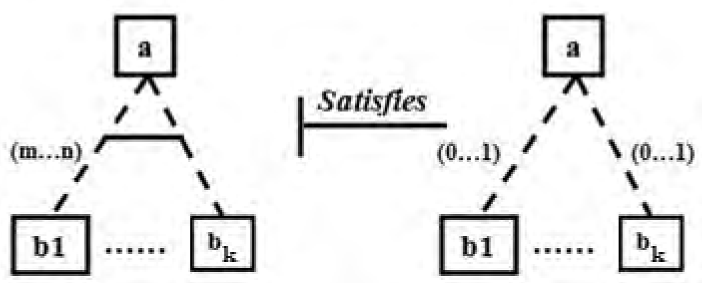

Fig. 10 Group feature (P) satisfying optional (P) features.

Group feature to Group feature (P): Group feature (P) satisfies Group feature (P), as shown in Fig.11.

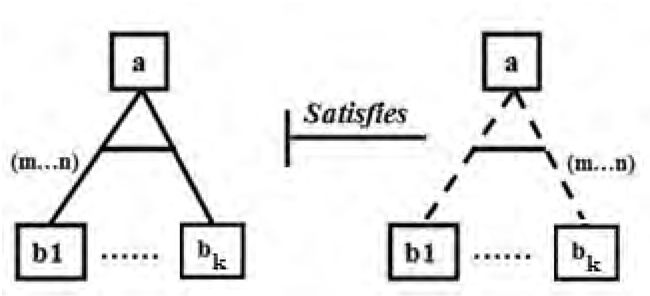

Fig. 11 Group feature (P) satisfying group feature (P).

5 Conclusion

The concept of cardinality based service feature diagram was proposed by Assad et al. (2015). This paper is the first step toward the matching and merging of cardinality based service feature diagram. In this paper we proposed graphical matching rule for cardinality based service feature diagram both for requestor and provider base.

References

Assad GM, Naeem M, Wahab HA. 2015. Towards cardinality-based service feature diagram. Computational Ecology and Software, 5(1): 69-76

Batory D. 2005. Feature models, grammars, and propositional formulas. In: Proceedings of SPLC’05, LNCS 3714. 7-20. Springer–Verlag, Netherlands

Czarnecki K, Eisenecker UW. 2000. Generative Programming: Methods, Tools and Applications. ACM Press/Addison-Wesley Publishing Co, New York, USA

Czarnecki K, Helsen S, Eisenecker UW. 2005. Formalizing cardinality-based feature models and their specialization. Software Process: Improvement and Practice, 10(1): 7-29

Czarnecki K, Wasowski A. 2007. Feature diagrams and logics: There and back again. In: Proceedings of SPLC’07. 23-34

Erl T. 2009. SOA Design Patterns. Prentice Hall PTR, USA

Gunter A, Kuhn C. 1999. Knowledge-based configuration: Survey and future directions. In: Proceedings of the 5th Biannual German Conference on Knowledge-Based Systems: Knowledge-Based Systems - Survey and Future Directions. 47-66

Heckel R, Cherchago A. 2007. Structural and behavioral compatibility of graphical service specifications. JLAP, 70(1.1): 15-33

Lee J, Muthig D, Naab M. 2008. An approach for developing service oriented product lines. In; Proceedings of SPLC’08. 275-284

Kang KC, Cohen SG, Hess JA, Novak WE, Peterson AS. 1990. Feature-oriented domain analysis (FODA) feasibility study. Technical Report, Carnegie-Mellon University Software Engineering Institute, USA

Kang KC, Kim S, Lee J, et al. 1998. Form: A feature-oriented reuse method with domain-specific reference architectures. Annals of Software Engineering, 5: 143-168

Naeem M, Heckel R. 2009. Flexible services specification and matching based on feature models. In: Proceedings of MDPLE’09 co-located with ECMDA-FA’2009. 36-44

Naeem M, Heckel R, Orejas F, Hermann F. 2010. Incremental service composition based on partial matching of visual contracts. In: Proceedings of FASE’10, LNCS 6013. 123-138, Springer, Netherlands

Narendra NC, Ponnalagu K, Srivastava B, Banavar GS. 2008. Variation-oriented engineering (VOE): Enhancing reusability of soa-based solutions. IEEE Conference on Services Computing, 1: 257-264

Rao J, Kungas P, Matskin M. 2006. Composition of semantic web services using linear logic theorem proving. Information Systems, 31: 340-360