

Connecting Feature Models and Ontologies in Software Product Lines

Gabriella Castro Barbosa Costa, Regina Braga, José Maria N. David and Fernanda Campos

Department of Computer Science, UFJF, Juiz de Fora, Brazil

Keywords: Product Line, Feature Model, Ontology, Domain Engineering.

Abstract: Feature models have been used as a way to describe the variability in Software Product Lines (SPL). Using this model we can show variation points, variants and restrictions between variation points and variants. However, restrictions and relations that can be expressed using this domain modelling technique are unsatisfactory. Our proposal aims to solve this challenge connecting feature models and ontologies as domain modelling technique, which can be used to provide additional information and can also add more semantics in the context of Software Product Lines.

1 INTRODUCTION

A key issue in Software Product Line Engineering (SPLE) is the specification of the domain model, which should express the variability of the SPL. To do this, we can use domain models like feature models and ontologies, for example. Using feature models we can show variation points, variants and restrictions between variation points and variants. However, restrictions and relations that can be expressed using this technique of domain modelling are unsatisfactory, considering the lack of semantics between the entities which composes the model as well as some restrictions associated with the terms.

The purpose of this paper is to present two domain modeling techniques: feature modeling and ontology modeling which could be combined to obtain the advantages of both and, as a result, to improve the SPL domain specification. In this work, we explore the ontologies to get the main restrictions that need to be applied to our SPL. These restrictions are used to improve the choice of features related to the product that will be developed. On the other hand, the feature model is used to facilitate the understanding about the features that need or not to be selected according to the product that will be developed.

In Section 2, we describe the essence of feature models and ontologies. In Section 3, three studies that use feature modeling, ontologies or both of them in SPLs are presented: (Czarnecki et al., 2006), (Johansen et al., 2010) and (Asikainen et al., 2007). In the next section, our approach to connecting feature models and ontologies in SPL is explained and an ex-

ample is shown. Conclusions about this work are presented in last section.

2 ESSENCE OF FEATURE MODELS AND ONTOLOGIES

Feature models show the variation points and the variants of a SPL. This model should also include the restrictions between the variation points and the variants, as a variation point (or a variant) may require or exclude a variation point (or a variant). However, feature models do not have a high power of description as ontologies. For example, using only features models there is no way to know when a particular feature is more suitable to be chosen than another. This lack of semantics in features models can be supplied through the possibility of inference offered by ontologies.

In figure 1, when can see this lack of semantics of a feature model. For example: using a feature model we can make a restriction that shows that a 'pairwise_local_aligning' is performed by an algorithm called BLAST, but we can not express which BLAST program is more appropriate for a particular type of 'pairwise_local_aligning'. Still according to figure 1, we can see that the task 'multiple_global_aligning' can be performed by two different algorithms. But, only through this feature model, we can not express when the algorithm 'MultiAlign' is more suitable than the 'CLUSTAL.W' algorithm.

Ontology modeling is another domain modeling technique. Ontology is "a formal and explicit specifi-

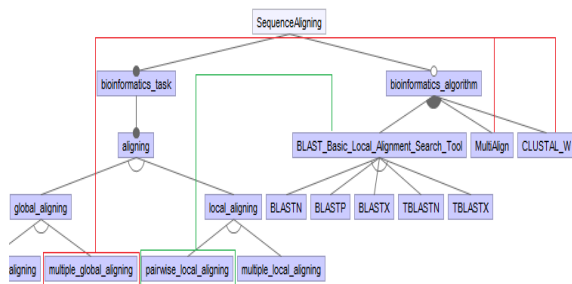


Figure 1: The lack of semantics of a feature model.

cation of a shared conceptualization” (Gruber, 1993).

Assuming that ontologies can be used to model a specific domain, it can also improve feature modeling by providing additional and relevant information for the domain of the SPL in which a feature model is constructed. On the other hand, we need the feature models because they are a simple way to understand and to represent the features in the SPL.

3 RELATED WORK

(Czarnecki et al., 2006) explore the relationship between feature models and ontologies and suggest the addition of extensions to the feature model in order to move it closer to richer formalisms for specifying ontologies. (Czarnecki et al., 2006) also explores the idea of the feature models as views on ontologies. In order to solve the gap between feature models and ontologies, our work suggests the combined use of feature models and ontologies. Differently from (Czarnecki et al., 2006), we do not show how to improve the feature model adding extensions on it. We also do not use ontologies to align the view between the feature models or outlines of requirements. Our focus is to use a feature model to represent the variability of the SPL and the ontology to provide the restrictions about the SPL that could not be represented only by the features models. Moreover, we propose the use of ontology and the possibility of making inferences in it to improve the choices in the SPL.

(Johansen et al., 2010) explore the synergies between feature models and ontologies, showing mapping mechanisms that a user can establish and take advantage of the relationship between feature models and ontologies. They explore the gap between these models and show scenarios where ontologies can be applied for feature modeling. Our proposal also suggests the use of ontologies and feature models to solve the gap between these models, like (Johansen et al., 2010). Differently, we focus on mapping the ontology classes and features of the feature model, getting

the restrictions about the domain using the ontology. After that, we show the features that need to be selected on the feature model, at runtime.

In (Asikainen et al., 2007), a domain ontology for modeling the variability in SPL, called Kumbang was proposed. This ontology unifies the feature modeling and the architecture modeling in SPL. Although (Asikainen et al., 2007) uses ontology to represent the features and the architecture of SPL, it should be considered that Kumbang ontology was described using UML. As a result, the power of the semantics description of this language is not the same as OWL languages, considering also the inference mechanisms that can be used in conjunction with OWL. In our work, we use both a domain ontology (expressed using OWL-DL) and a feature model to support domain modeling in a SPL.

4 CONNECTING FEATURE MODELS AND ONTOLOGIES

We propose the development of a SPL to support scientific applications construction in Biology, in order to tackle the sequence alignment process problem. We choose this domain to apply the proposed approach because our group has previous works which involves bioinformatics, scientific software and use of ontologies, as can be seen in (Matos et al., 2010), (Mendes et al., 2011) and (Silva et al., 2012).

The initial architecture of the SPL is being detailed considering its main modules (called managers), as can be seen in Figure 2: (i) the Product Line Manager, which interacts with the user and the other managers in the architecture, coordinating their behavior, (ii) the Domain Model Manager (which includes the Variability Manager) is responsible for maintaining the domain models in accordance with the SPL scope, (iii) the Variability Manager, which is responsible for managing the models that represent the variability in the SPL and (iv) the Architectural Manager is responsible for the SPL architecture. These managers interact with artifacts repository, which should include the components and devices for the generation of the SPL products. To build SPL products, we can also access distributed repositories.

The sequence alignment process can perform the complete alignment of sequences (called global alignment) or only compare fragments of the sequences (called local alignment), as can be seen in Figure 3(a). The sequence alignment can also indicate the degree of similarity between two sequences (called pairwise alignment) or more than two sequences (called mul-

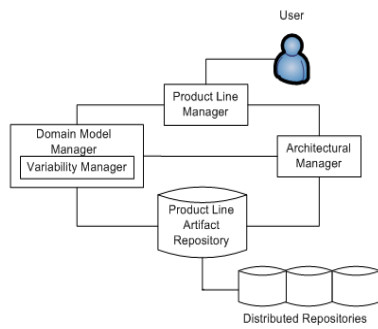


Figure 2: Product line Architecture.

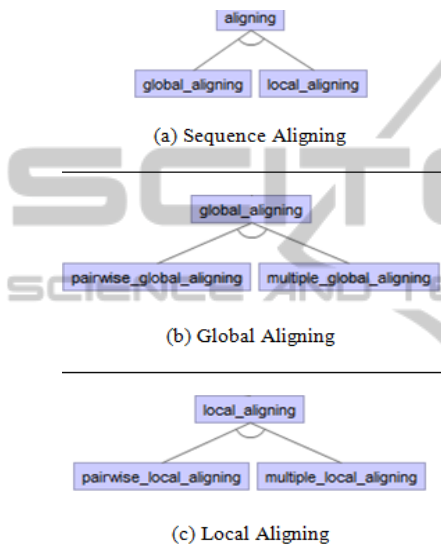


Figure 3: Global, local, pairwise and multiple aligning.

multiple alignments). Both local and global alignments can be performed with two or more sequences, as can be seen in Figures 3(b) e 3(c).

This paper focuses on 'pairwise local aligning', shown in Figure 3(c). The 'pairwise local aligning' can be executed by the family of algorithms known as BLAST (Basic Local Alignment Search Tool). This family can be divided into five different programs, as can be seen in Figure 4.

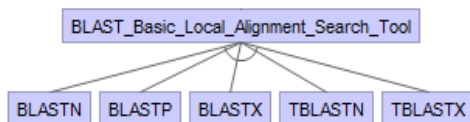


Figure 4: BLAST family of programs.

In our example, we aimed to define which of BLAST is more suitable for a specific type of 'pairwise local aligning'. To address this issue, we identified the main required parameters to perform each BLAST algorithm. These required parameters were represented as restrictions in our ontology. So, the

three following parameters were considered: (i) the input sequence type, (ii) the format of the compared sequence and (iii) the database used for the query.

Using only a feature model, we cannot express the semantic about our domain as we can do using ontologies. To solve this lack of semantics of our feature model, we use a Sequence Alignment Ontology. This ontology was adapted from the myGrid ontology (Wolstencroft et al., 2007). We use only part of it and add some extensions in order to make the ontology usable in conjunction with our feature model. It is important to emphasize that we have some work on this topic using other ontologies, mainly CeLO (Matos et al., 2010), (Silva et al., 2012), but in this example, the myGrid Ontology is more suitable.

In our approach, the Variability Manager is responsible for the connection between the models. This manager works according to the following steps:

1. It takes the mandatory features from the feature model and presents them to the user. So, the user can make the choice of a desired subfeature of the mandatory features.

2. The manager picks the features that were selected by the user and accesses a simple XML file. This file has a mapping between the feature model and the classes of the ontology. Currently, this XML file is created manually, with the help of a domain specialist. However, we are planning to use some ontology alignment techniques, tailored in order to be used with a feature model.

3. With the corresponding class name, the Variability Manager can access this class in the ontology and can get its restrictions.

4. As the restrictions refers to features that should be selected, once they are related to those features that were selected at an earlier time, the variability Manager consults the XML mapping and gets the feature name of the restriction.

5. Finally, the new set of features to be selected is displayed to the user, in accordance with the restrictions which were found.

These five steps should be performed until the configuration of the features necessary for a given product is completed and there are not more restrictions to be found.

It is important to notice that, in this process, the inference mechanism in the ontology can act in order to improve the selections. The use of inferences in ontologies can be considered one of the main advantages of this modeling technique domain and may be used in our approach to find other links between the different features available, in order to provide a better selection of the features in our SPL.

In our example, the mandatory features and its

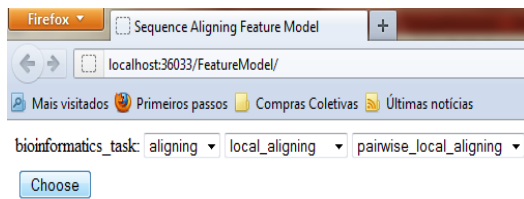


Figure 5: Feature selection.



Figure 6: Restriction of 'pairwise_local_aligning'.

subfeatures are shown to the user using selection boxes. As illustrated in Figure 5, the subfeature 'pairwise_local_aligning' was selected. After that, the Variability Manager uses the XML mapping file to obtain the class name of the feature 'pairwise_local_aligning'. It consults the restrictions of the class in the ontology. This class has two restrictions, as can be seen in Figure 6.

Using these restrictions, the Variability Manager consults the XML to transform the classes of the restrictions into features. Next, in our example, is shown to the user a selection box with the subfeatures of 'bioinformatics_algorithm' and an information to select the feature 'BLAST_Basic_Local_Alignment_Search_Tool'.

When the BLAST bioinformatics algorithm is selected, the Variability Manager repeats the step 2, as was proposed before, and obtains the restrictions of the BLAST. According to the BLAST restrictions, the user must select: (i) the format of the input sequence, (ii) the format of the sequence to be compared with and (iii) the format of the sequence database. Finally, after the selection of this features, the Variability Manager has to associate all the selected features and its restrictions. After that, it needs to find, in the ontology, the most appropriate class that has these restrictions.

For example, if the user selects: (i) a protein input sequence, (ii) a protein sequence format to compare with and (iii) a database with sequences of protein, the Variability Manager can inform, through the ontology analysis, that BLASTP is the most compatible class with the previously selected features.

It should be emphasized that this work is only the first step towards the proposed SPL. Our research encompasses all the Sequence Alignment SPL.

5 CONCLUSIONS

Through this paper, we present an approach to con-

nect a feature model and ontology using a Variability Manager and a XML mapping file. It aims to obtain the advantages of both domain modeling techniques. To illustrate our approach we applied this approach in the bioinformatics context, attacking the sequence aligning problem. Through our example, we illustrated how an ontology can improve the feature model, providing additional and relevant information for the domain of the proposed SPL.

For further work, the issues below remain to be resolved in this research: (i) improve and evolve our models, incorporating into this all the tasks needed in a Sequence Alignment SPL; (ii) generate a semi-automatic matching between the feature model and the ontology and (iii) improve the application interface by which the user must select the features.

REFERENCES

- Asikainen, T., Männistö, T., and Soinen, T. (2007). Kumbang: A domain ontology for modelling variability in software product families. *Adv. Eng. Inform.*, 21.
- Czarnecki, K., Peter Kim, C. H., and Kalleberg, K. T. (2006). Feature models are views on ontologies. In *Proceedings of the 10th International on Software Product Line Conference*, pages 41–51, Washington, DC, USA. IEEE Computer Society.
- Gruber, T. R. (1993). Toward principles for the design of ontologies used for knowledge sharing. In *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Kluwer Academic Publishers.
- Johansen, M., Fleurey, F., Acher, M., Collet, P., and Lahire, P. (2010). Exploring the synergies between feature models and ontologies. In *International Workshop on Model-driven Approaches in Software Product Line Engineering (MAPLE 2010)(SPLC'10 (Volume 2))*, volume 2 of *SPLC'10 (Volume 2)*, pages 163–171, Jeju Island, South Korea. Lancaster University.
- Matos, E. E., Campos, F., Braga, R., and Palazzi, D. (2010). Celows: An ontology based framework for the provision of semantic web services related to biological models. *J. of Biomedical Informatics*, 43:125–136.
- Mendes, L. F., Silva, L., Matos, E., Braga, R., and Campos, F. (2011). Sasagent: An agent based architecture for search, retrieval and composition of scientific models. *Comput. Biol. Med.*, 41:449–462.
- Silva, L. M., Braga, R., and Campos, F. (2012). Composerscience: A semantic service based framework for workflow composition in e-science projects. *Inf. Sci.*, 186:186–208.
- Wolstencroft, K., Alper, P., Hull, D., Wroe, C., Lord, P. W., Stevens, R. D., and Goble, C. A. (2007). The mygrid ontology: bioinformatics service discovery. *Int. J. Bioinformatics Res. Appl.*, 3:303–325.