Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 18 (2013) 759 – 768

Procedia Computer Science

2013 International Conference on Computational Science

PL-Science: A Scientific Software Product Line

Gabriella Castro B. Costa[a], Regina Braga[a], José Maria N. David[a], Fernanda Campos[a], Wagner Arbex[b]

[a]*Federal University of Juiz de Fora – Department of Computer Science, Juiz de Fora, Brazil*
[b]*Brazilian Agricultural Research Corporation - National Research Centre of Dairy Cattle, Juiz de Fora, Brazil*

Abstract

A way to improve reusability and maintainability of a family of software products is through the use of Software Product Line (SPL) approach. Software families, also named SPLs, are a set of software intensive systems sharing a common set of features which are managed to satisfy specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way. This paper presents the PL-Science approach that considers the context of SPL and aims to assist scientists to define a scientific experiment, specifying a workflow that encompasses scientific applications of a given experiment. Using SPL concepts, scientists can reuse models that specify the scientific product line, and carefully can make decisions according to their needs. In the context of this paper, Scientific Software Product Lines (SSPL) differs from the Software Product Lines (SPL) due to the fact that SSPL uses an abstract scientific workflow model. This workflow is defined according to a scientific domain and, using this abstract workflow model, the products (scientific applications/algorithms) will be instantiated. This paper also focuses on the use of ontologies to facilitate the process of applying Software Product Line (SPL) to scientific domains. Through the use of ontology as a domain model, we can provide additional information as well as add more semantics in the context of Scientific Software Product Lines (SSPL).

Keywords: Product Line; Ontology; Feature Model; Scientific Workflows; Sequence Alignment.

1. Introduction

Software families, also named Software Product Line (SPL), are "a set of software intensive systems sharing a common set of features which are managed to satisfy specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way" [5]. When the SPL approach is used, it is necessary to emphasize the importance of the domain model specification. Through domain models, we can express SPL variabilities which differentiate a given application from another in the same domain. To represent these variabilities, we can use domain models such as, feature models, ontological models or profiled UML class diagrams. Each of these models has its advantages and disadvantages to support variability and

doi:10.1016/j.procs.2013.05.240

commonality representations. Researches such as [2], [6-7] and [10] provide approaches that use ontologies to improve SPLs. In these studies, one problem is recurrent: the need to add more semantics in SPL variability representation. The purpose of this paper is to present a way to improve SSPL (Scientific Software Product Lines) domain specification using ontologies in addition to feature models, considering the scientific context and its specificities. As a result, we could obtain the advantages of these two domain model techniques to generate scientific workflows through an SPL approach. As will be described in this paper, we want to extract the best of both model types, i.e., the feature model will be used to support variability representation and the ontology will be used to express formal restrictions and possible inferences on these restrictions, considering that the scientific domain needs a formal specification, as we will explain latter. The 'alignment' between these models is enriched because we try to extract the semantics of both, improving the SSPL knowledge base.

The remainder of the paper is organized as follows. In Section 2, is presented an overview of PL-Science approach, presenting the main models, proposed architecture, and methodology. Section 3 shows an example of applying the approach in bioinformatics domain. Section 4 presents related works. Finally, section 5 presents the conclusions and future work in order to improve the PL-Science approach.

2. PL-Science Overview

Our approach, named PL-Science, considers the context of Software Product Line and aims to assist scientists to define a scientific experiment, through the specification of a workflow that encompasses scientific applications of a given experiment. Using SPL concepts, scientists can reuse the models that specify the scientific product line and carefully can make decisions according to their experiment.

In the context of this paper, Scientific Software Product Lines (SSPL) differs from the Software Product Lines (SPL) due to the fact that SSPL uses an abstract scientific workflow model (Figure 1) and also uses a domain ontology that formally specify the scientific domain, including its formal restrictions. This workflow is defined according to this scientific domain and, using the abstract workflow model, the products (scientific applications) will be instantiated on this workflow. It is also important to note that using PL-Science, a scientist can specify one or more isolated scientific application, without a scientific workflow.

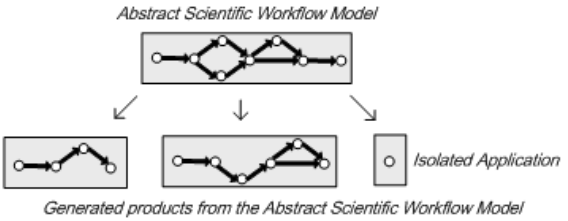

Figure 1. Abstract Scientific Workflow Model

This approach focuses on requirements engineering stage in SSPL, considering the domain analysis of a SPL. Analyzing the difficulties in specifying scientific experiments, and considering the need to compose scientific applications in order to execute an in virtuo scientific experiment, more appropriate semantic to support the domain analysis stage is needed. Thus, this paper proposes the connection of feature models and ontologies in order to combine their benefits to the domain modeling. The hypothesis is that by the use of feature model associated to ontology, we can use the semantic knowledge provided by ontologies in order to facilitate the selection and organization of scientific workflows in SSPL context. We also have as ultimate goal the generation of scientific workflows with the activities to be carried out according to the chosen domain and user needs, or the suggestion of scientific applications relevant to a given scientific task, considering an in virtuo experiment context.

In SSPL context, the domain understanding and the selection of desirable features in a workflow or in a scientific application is a key issue, since it is very difficult to scientists develop this type of application. One way to simplify this process is to provide more understanding to the scientist. Thus, we present the details of PL-Science domain models (feature model, ontology and mapping file), considering as the target domain, the genetic sequencing domain. In SPL, one of the first activities to be performed is the feature analysis, identifying the externally visible characteristics about the products of the SPL and organizing them in a feature model. This model shows the variation points (where the characteristics of the product line may vary) and the variants (possible values of a point of variation) of an SPL. It 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). In order to provide more semantics to SSPL, we tackled the problem of what kind of domain model to be used. One possibility would be feature models. However, in this approach, the use of feature models alone seemed to be limited. Through these models we cannot express all the restrictions needed in the scientific domain (in our case, in the genetic sequencing and alignment domain). Other authors have also pointed out this difficulty. According to [7], feature models were not designed to enhance interoperability, information retrieval and inference. This fact is also emphasized by [6] and [10]. Feature models do not offer, for example, the possibility to express all the semantics which is involved in relationships between features needed in scientific applications. As an example, as can be seen in Figure 2, using only features models, we can establish some relationships between the features. For example, we can say that the selection of the 'pairwise_local_aligning' feature implies the selection of BLAST feature, or, the selection of the 'sequence_grouping' feature implies the selection of one of the features 'CAP3' or 'Phrap'. However, using only feature models we cannot express when the selection of a particular feature becomes more appropriate than another. For example, how to express when the selection of the feature CAP3 is more appropriate than the selection of the features 'Phrap' (in the case of feature 'sequence_grouping' have been selected)?

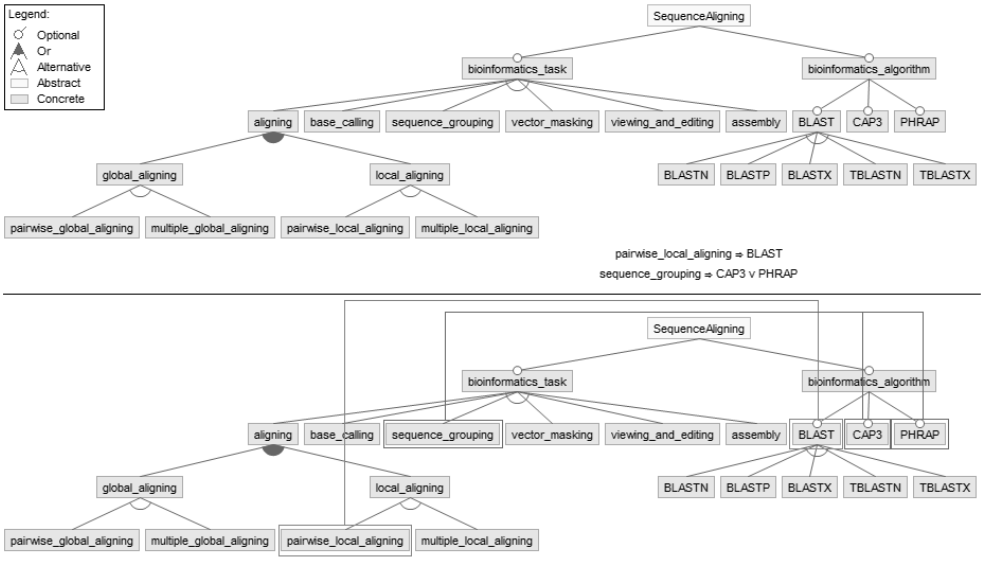

Figure 2. Feature Model

In our approach, this lack of semantics in features models can be supplied through ontologies. Another advantage of ontologies is the possibility of using inference mechanisms. Therefore, we have decided to connect features models to ontologies in order to enhance the semantics in product specifications (scientific

workflows) to be generated by the SPL. It should be highlighted the possibility to express association rules and dependencies between features through assertions, described by the propositional logic, as was done by other authors [8], for example. Through this issue, we can add more semantics to the feature model. But in our case, the use of ontologies (described in OWL) to formalize the restrictions of the variations points in the SPL allows to represent knowledge that would not be possible using only propositional logic. Moreover, the restrictions creation using OWL is much simpler than propositional logic utilization. Ontology is a formal and explicit specification of a shared conceptualization. It allows capturing the common understanding of objects and their relationships in a particular domain [9]. Assuming that ontologies can be used to model a specific domain, it can also improve feature modeling providing additional information to the domain of the SPL to generate scientific workflows. It is important to consider that our research group has other previous works which have involved the use of ontologies and scientific software, as can be seen in [11], [12] and [13].

One of the most important characteristics of ontologies is the possibility of using inference engines (*reasoner*) to obtain new knowledge which is not explicit. In this context, an inference machine can "infer" a new hierarchy in accordance with what was defined in the ontology. Thus, the inference engine can be used to test whether a given class is a subclass of another class declared in the ontology. Another advantage offered by inference engines is the ontology consistency checking. Figures 3 and 4 show an inference example. In these figures there is a small part of the Sequence Alignment Ontology, used for implementing PL-Science (it is important to note that the Sequence Alignment Ontology used in this work was adapted from the MyGrid Ontology [14]; we use part of it and add some extensions in order to make the ontology more usable in conjunction with the feature model).

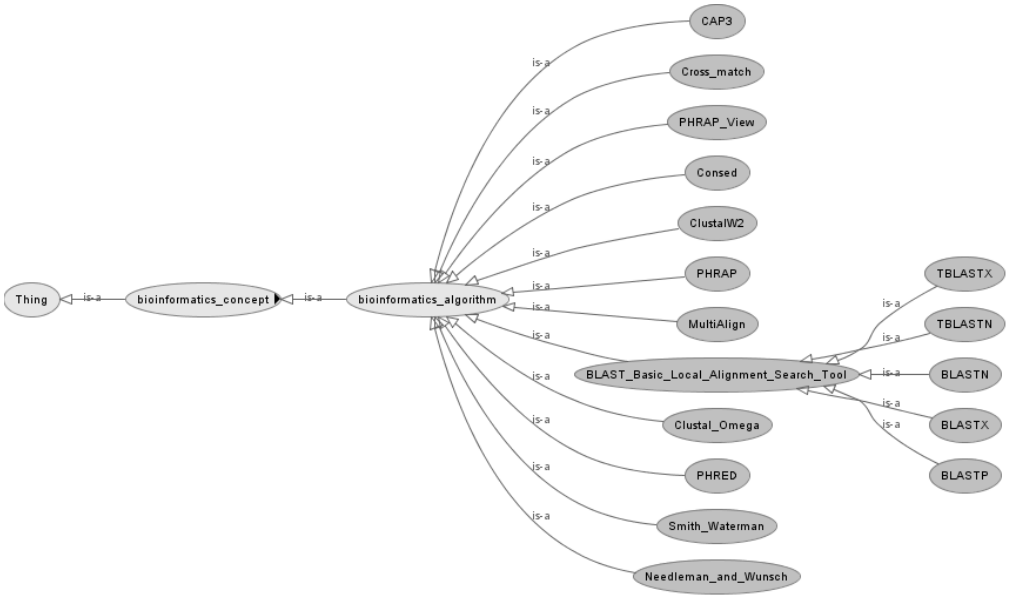

Figure 3. Asserted Model

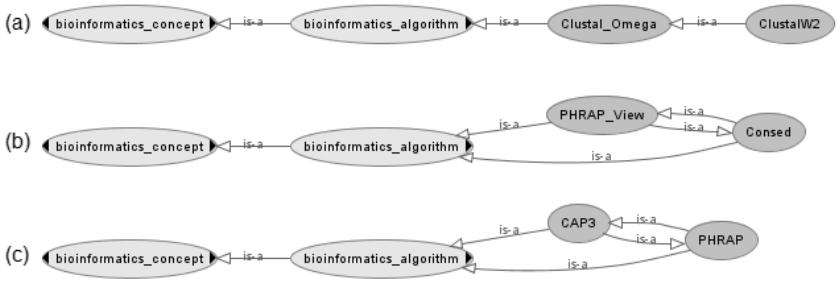

Figure 4. Inferred Model

In Figure 3, there is a class hierarchy as they were declared. It can be highlighted from this figure that the classes 'ClustalW2' and 'Clustal_Omega' are subclasses from the 'bioinformatics_algoritms' class and occupy the same position hierarchically. This same situation occurs with the 'PHRAP_View' and 'Consed' classes, and also with the 'PHRAP' and 'CAP3' classes. After processing the inference algorithm, it is possible to visualize what is shown in Figure 4, i. e., according to the existing constraints between classes, we can visualize that the 'ClustalW2' becomes a subclass of 'Clustal_Omega' (Figure 4(a)). In the Figure 4(b), we have that 'PHRAP_View' and 'Consed'classes are similar. The same algorithm applies to the classes shown in Figure 4(c). Considering these restrictions and the importance of supporting scientists in the task of defining and implementing new applications, this paper proposes the use of two domain models (feature models and ontologies). When combined, these models can bring more expressiveness and, as a result, facilitate the feature selection to develop workflows/scientific applications.

2.1. PL-Science Architecture

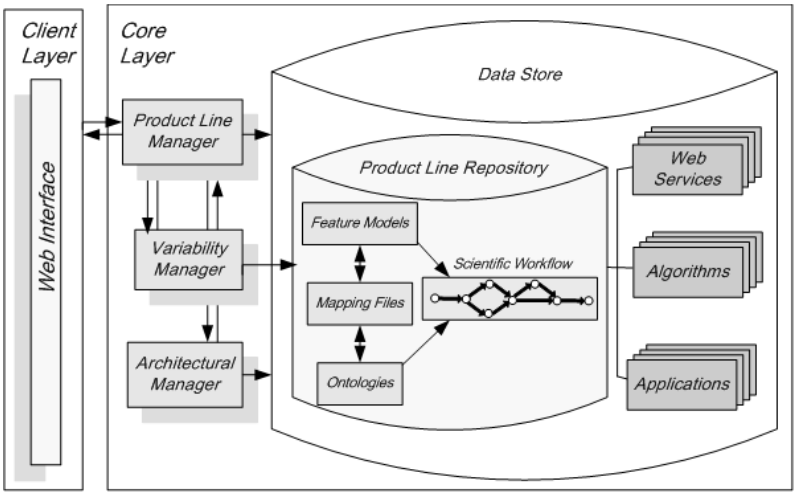

Figure 5. PL-Science Architecture

According to Figure 5, PL-Science architecture is divided into two main layers: Client Layer, which includes the web interface of the application and the Core Layer. The latter is composed by three managers: Product Line Manager, Architectural Manager and Variability Manager. These three managers interact with the Software Product Line Artifact Repository, which includes the components to support the scientific workflows

generation. The SPL Artifact Repository contains the feature model that describes the variability of the SPL domain, and the ontology, which describes the formal semantics of the domain. To connect these two models, we use a Mapping File. Finally, the SPL repository should also include the algorithms, web services and applications. These artifacts will be used to compose the scientific workflow, according to the scientist requirements.

3. Case Study

This section presents a use of PL-Science approach based on Bioinformatics domain, to generate a scientific workflow for the genetic sequencing/alignment process. Considering the benefits of SPL in the context of applications that share a core of common artifacts, we apply the proposed approach to support scientific experiments in gene sequencing domain [3]. Analyzing the difficulties in specifying scientific experiments, and considering the composition of scientific applications for their implementation, a more appropriate semantic support for the domain analysis stage is needed. The hypothesis is that using a feature model associated to ontological models, we can facilitate the selection and structuring of scientific workflows in the context of an SSPL. Thus, the goal of this research is to provide the application of the proposed approach for SSPL, and then, analyze the benefits derived from the approach.

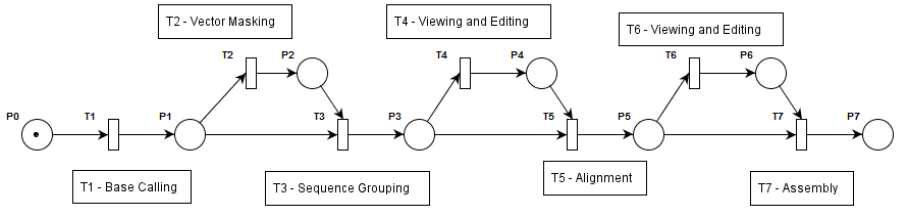

Figure 6. Abstract Workflow Model

PL-Science proposes a methodology for SSPL development. It has two main phases, Core Artifacts Development and Product Development. According to the proposed methodology and considering the sequence alignment domain, the steps are:

- *Phase 1*: Core Artifacts Development: **Step 1:** Defining the scope of the scientific software product line; **Step 2**: Defining the feature model in accordance with the existing possibilities of variation in the chosen domain; **Step 3**: Defining the domain ontology, containing the 'restrictions' of the selected domain that cannot be expressed only by the features model; **Step 4**: Creating the mapping file between the terms of the feature model and the domain ontology; **Step 5**: Defining an abstract workflow model, according to the chosen domain; **Step 6**: Defining and connecting other artifacts (algorithms, Web Services or applications) that will be stored in the SSPL repository.

Step 1: The scope of the SPL is related to scientific workflow that encompasses the activities involved in the sequence alignment process, which are: base calling, vector masking, sequence grouping, visualization and edition, alignment and assembly; **Step 2**: In this step, the feature model was defined. Figure 2 shows a small part of the defined feature model. The complete model can be seen at http://gabriellacastro.com.br/PL-Science/SequenceAligningFeatureModel.xml; **Step 3**: The domain ontology was developed. As previously mentioned, it was derived from the MyGrid ontology. The complete ontology can be seen at http://gabriellacastro.com.br/PL-Science/SequenceAligningOntology.owl ; **Step 4**: In this step, a file that maps the terms in the feature model into the terms of the ontology (and vice-versa) was created. The complete mapping file is available at http://gabriellacastro.com.br/PL-

Science/SequenceAligningMapping.xml; **Step 5**: The abstract workflow model developed for the chosen domain is illustrated in Figure 6. This figure is only a visual representation of the workflow. All the possibilities about the sequence of the tasks that this workflow encompasses were described through the specified domain ontology; **Step 6**: In this step, several web services, algorithms, or other types of application have been cataloged as SSPL artifacts. These artifacts will be used later to compose the scientific workflow. For example, we have catalogued various web services from BioCatalogue [4], such as 'runPhrapService' (http://www.biocatalogue.org/services/1567) and 'INB-dev:genome.imim.es:runPhrap' (http://www.biocatalogue.org/services/2268).

- *Phase 2: Product Development using the Core Artifacts:* **Step 1**: Available features in the chosen domain are selected according to the products to be developed (scientific workflow or isolated application). This selection will be based both on feature model and their mapping into the ontological model, as well as in the workflow(s) model(s) available for the domain. In this step, the scope of SPL should also be taken into account. This step is controlled by the *Variability Manager*; **Step 2**: For each task in the base workflow, the possibilities of variation (defined in step 2) is analyzed. After that, the user can define which algorithm, Web service or application will be 'instantiated' in the base workflow; **Step 3**: A XML file is generated which details the tasks that will compose the scientific workflow (or an isolated application).

  Step 1: To support the feature selection, we developed a web application. Its home page is shown in Figure 7. At the first time, mandatory features, which were established in feature model, are shown to the user to be selected as an associated subfeature. In the chosen domain, these mandatory features are the 'type of application' to be developed and the 'sequence platform'. The first feature to select was the application type to be created. In this example, we have chosen the feature 'workflow' (another possibility is an 'isolated application'). The restrictions about these features are shown in Figure 8 ((a) shows the restrictions about 'isolated application' and (b) the restrictions about 'workflow'). After that, the user needs to select the sequence platform of sequencing. So, the Variabilty Manager (that analyses the restrictions of selected features) informs to the Product Line Manager that the next feature to be selected is the tasks of our workflow. In this case study, we want to create a pipeline (base workflow) based on [1], that involves the definition of a workflow which is capable of performing the tasks T1 (base calling), T2 (vector masking) and T3 (sequence grouping) of the Abstract Workflow Model (Figure 6). So, in the feature selection we choose the final task 'sequence_grouping'; **Step 2**: For each task (T1, T2, T3) the Variability Manager analyzes the possibilities of variations. After that, the user can define which algorithm, Web service or application will be 'instantiated' in the base workflow for each task. For example: the task T3, through an ontology searching, can be performed by algorithms (or web services) PHRAP or CAP3. Using the inference mechanism, we confirm this fact, as can be seen in Figure 4(c). After that, were shown to the user all the possibilities of services and algorithms of the type PHRAP and CAP3 and he/she selects the most appropriate to compose the workflow. At this point, we can highlight another advantage of ontologies. Through the use of them, we can define a restriction for the 'Phrap'class that is most appropriate to be selected when an algorithm that belongs to class 'Phred' was used for the task of 'base_calling'. The identification of this restriction was not possible using only feature models. For example, it was shown to the user the possibility to select 'runPhrapService' (http://www.biocatalogue.org/services/1567) or to use a web application as 'CAP3 Sequence Assembly Program' (http://pbil.univ-lyon1.fr/cap3.php), and a message that 'PHRAP' services type are more suitable if he/she had selected a 'PHRED' service before to solve the task of 'base_calling'. These are just examples for the task of 'sequence_grouping', but the application shows all the possibilities that meet the user selected task. It is important to note that all the tasks that will compose the workflow are

analyzed and the inference mechanism is used to find out all the possibilities types of 'bioinformatics_algoritms' that can be used to solve the selected task; **Step 3**: A XML file was created, containing details of the scientific workflow tasks. The XML file of the example is shown in Figure 9.

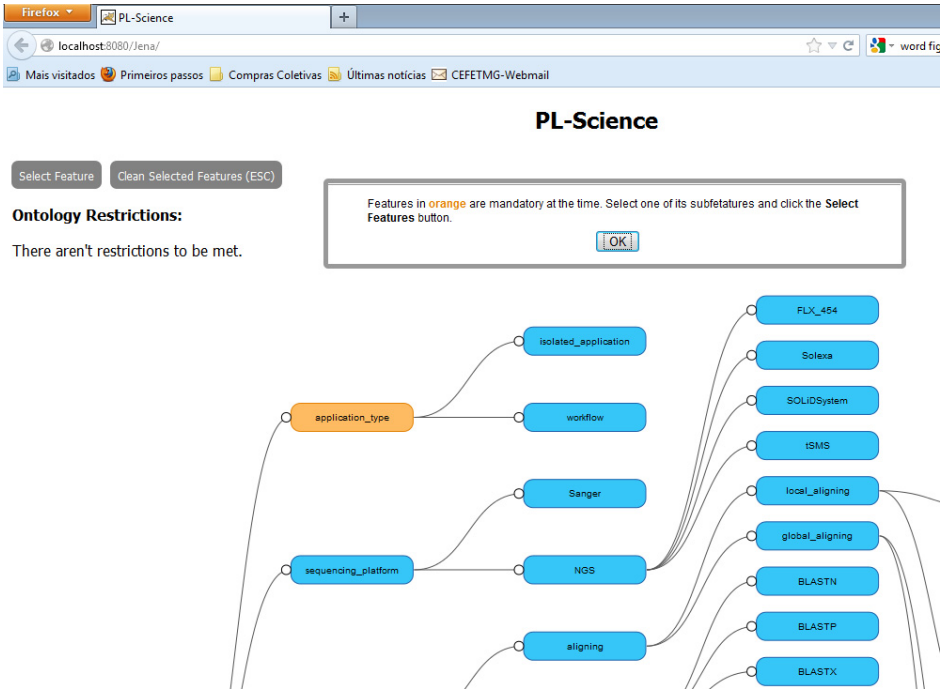

Figure 7. Feature Selection Application

(a)

```
<!-- http://gabriellacastro.com.br/SequenceAligningOntology#isolated_application -->

<owl:Class rdf:about="http://gabriellacastro.com.br/SequenceAligningOntology#isolated_application">
    <rdfs:subClassOf rdf:resource="http://gabriellacastro.com.br/SequenceAligningOntology#application_type"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="http://gabriellacastro.com.br/SequenceAligningOntology#workflow_possible_tasks"/>
            <owl:onClass rdf:resource="http://gabriellacastro.com.br/SequenceAligningOntology#bioinformatics_task"/>
            <owl:qualifiedCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:qualifiedCardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
```

(b)

```
<!-- http://gabriellacastro.com.br/SequenceAligningOntology#workflow -->

<owl:Class rdf:about="http://gabriellacastro.com.br/SequenceAligningOntology#workflow">
    <rdfs:subClassOf rdf:resource="http://gabriellacastro.com.br/SequenceAligningOntology#application_type"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="http://gabriellacastro.com.br/SequenceAligningOntology#workflow_possible_tasks"/>
            <owl:someValuesFrom rdf:resource="http://gabriellacastro.com.br/SequenceAligningOntology#bioinformatics_task"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
```

Figure 8. Feature´s Restrictions

```
<?xml version="1.0" encoding="UTF-8"?>
<Workflow>
  <Processes>
    <Process name="INB-dev:genome.imim.es:runPhred" id="Process_1">
      <Parameters>
        <Parameter name="algorithm_type" value="phred" config_value="true"/>
        <Parameter name="hasLocation" value="http://www.biocatalogue.org/services/2290"/>
      </Parameters>
    </Process>
    <Process name="cross_match" id="Process_2">
    <Process name="NB-dev:genome.imim.es:runPhrap" id="Process_3">
  </Processes>
  <Connections>
    <Connection name="Connection_1" source="Process_1" target=" Process_2"/>
    <Connection name="Connection_2" source="Process_2" target="Process_3"/>
  </Connections>
</Workflow>
```

Figure 9. Workflow Description

4. Related Work

In [6] is presented the relationship between feature models and ontologies. In this work, features models are also shown as views on ontologies. They identify different mapping patterns and show how these mappings can be specified using configurable Object Constraint Language (OCL) constraints. In order to solve the gap between feature models and ontologies, our work suggests the combined use of feature models and ontologies. Differently from [6], we do not show how to improve the feature model adding extensions on it. We also do not use ontologies to align views between the feature models or outlines of requirements. Our focus is on the use of a feature model to represent SPL variability and ontology to provide the restrictions about the SPL that could not be represented by the features models alone. It is very difficult to use only OCL to express the restrictions between the features on features models. Based on this argument, we propose the use of ontology and the possibility of making inferences to improve the choices of variabilities in the SPL.

In [10] the synergies between feature models and ontologies are also explored, showing mapping mechanisms that a user can establish and take advantage of the relationship between one or more feature models and one or more ontologies. Our proposal also suggests the association of ontologies and feature models to solve the gap between these models. Differently, we focus on mapping the classes of the ontology 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 [2], domain ontology for modeling variability in software product families, named Kumbang, was proposed. This ontology unifies the feature modeling and the architecture modeling in software product families. The Kumbang ontology was described using both UML (Unified Modeling Language) 2.0 profile and a natural language. In our work, we use both domain ontology (expressed using OWL-DL and not UML) and a feature model to support domain modeling in an SPL.

In [7], methods for adding domain information and descriptions of variability, using an upper ontology that specifies generic concepts and relationships in SPL are proposed. This approach reuses the SPL feature model adding semantic descriptions to it. However, it does not modify the notation of feature model. In [7], in order to enhance the semantics of an SPL, in a first moment, the feature model is mapped automatically to an ontology. Comparing [7] with PL-Science approach, we have that [7] has not been developed for a specific domain, it is generic. PL-Science approach focuses on the development of scientific software product line. Therefore, it is noteworthy that the scientific software field requires a greater degree of formalism for specifying constraints, especially if we consider that certain scientific experiments may contain processes that involve risks such as human life, for example. So, feature model provides a representation of SPL variability and ontology is used to represent the semantics of the domain that is not covered by the feature model. Thus, two models are used, without the need to modify 'the formats' of either.

5. Conclusion and Further Work

Four studies ([2],[6],[7] and [10]) that use feature modeling, ontologies or both of them in SPLs were presented in the related work section. In these studies, it is evident the need to add more semantics in the variability representation of a scientific workflow SPL. This paper presented an approach to connect feature model and ontology in order to construct a Scientific Software Product Line. It aims to obtain the advantages of both techniques. We also presented an example using the proposed approach. This example highlighted the utility of the restrictions expressed through the ontology and the advantages of inference mechanism.

At first, to use the approach, the scientist need to define four domain models (feature model, ontology, mapping between feature models and ontology and the workflow abstract model) as well as other artifacts, like algorithms, Web Services or other applications that will support the final workflow composition. After that, all these models will be used during the application engineering phase. One advantage of the approach is: the ontology to be used may be based on already established domain ontology. As for approach automating improving, it should be considered the possibility of semi-automation for creating the file mapping between the ontology and the feature model. However, it is important to check the mapping file generated by the scientist. All other models, at this time of the survey, are not amenable to automation. In the application engineering phase of the SPL, the generation of products is mostly automated and only requires user interaction in order to establish the product features of the workflow (or isolated application) to be generated.

For further work, we need to develop the architectural manager, enabling the generation of configurable architectures using PL-Science and to improve the models and PL-Science application in order to provide its connection with a scientific workflow management systems, to generate workflows that can run in these tools.

References

[1] Arbex, W., 2009. Computational models for the identification of genomic information associated with resistance to the cattle tick. PhD Thesis. Ph.D. in Systems Engineering and Computer Science - Federal University of Rio de Janeiro (in portuguese).

[2] Asikainen, T. et al, 2007. Kumbang: A domain ontology for modelling variability in software product families. *In Advanced Engineering Informatics*, Vol. 21, No. 1, pp 23-40.

[3] Baxevanis A. D., Ouellette B. F. F., 2001. Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins. Ed. Wiley-interscience. 2 ed. 470p.

[4] Bhagat, J.; Tanoh, F.; Nzuobontane, E.; Laurent, T.; Orlowski, J.; Roos, M.; Wolstencroft, K.; Aleksejevs, S. et al., 2010. BioCatalogue: A universal catalogue of web services for the life sciences. Nucleic Acids Research, Vol. 38.

[5] Clements, P. and Northrop, L., 2002. Software Product Lines: Practices and Patterns. Addison-Wesley Publishing Company, Boston.

[6] Czarnecki, K. et al, 2006. Feature models are views on ontologies. Proceedings of the 10th International on Software Product Line Conference. Washington, DC, USA, pp. 41-51.

[7] Filho, J. B. F., Barais, O., Baudry, B., Viana, W., Andrade, R. M. C. 2012. An approach for semantic enrichment of software product lines. In Proceedings of the 16th International Software Product Line Conference , Vol. 2. ACM, New York, NY, USA, 188-195.

[8] Elfaki, A., Phon-Amnuaisuk, S., Ho, C.K., 2009. Modeling Variability in Software Product Line using First Order Logic, 7th The International Conference on Software Engineering Research, Management and Applications (SERA2009), Haikou, China.

[9] Gruber, T. R., 1995. Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies - Special issue: the role of formal ontology in the information technology, Vol. 43, No. 5-6, pp 907-928.

[10] Johansen, M. F. et al, 2010. Exploring the Synergies Between Feature Models and Ontologies. Proceedings of the 14th International Software Product Line Conference, Vol. 2. Lancester University, pp 163-171.

[11] Matos, E. E. et al, 2010. Celows: An ontology based framework for the provision of semantic web services related to biological models. Journal of Biomedical Informatics, Vol. 43, No. 1, pp 125-136.

[12] Mendes, L. F. et al, 2011. Sasagent: An agent based architecture for search, retrieval and composition of scientific models. Computers in Biology and Medicine, Vol. 41, No. 7, pp 449-462.

[13] Silva, L. M. et al, 2012. Composer-science: A semantic service based framework for workflow composition in e-science projects. In Information Sciences, Vol. 186, No. 1, pp 186-208.

[14] Wolstencroft, K. et al, 2007. The mygrid ontology: bioinformatics service discovery. International Journal of Bioinformatics Research and Applications, Vol. 3, No. 3, pp 303-325.