AUTOMATED ANALYSIS OF
SOFTWARE PRODUCT LINES

AN APPROACH TO DEAL WITH ORTHOGONAL
VARIABILITY MODELS

FABRICIA CARNEIRO ROOS
UNIVERSITY OF SEVILLE

THESIS PROJECT
ADvised BY DR. ANTONIO RUIZ CORTÉS AND DAVID BENAVIDES CUEVAS

JULY, 2009
First published in July 2009 by
The ISA Group
ETSI Informática
Avda. de la Reina Mercedes s/n
Sevilla, 41012. SPAIN

Copyright © MMIX The ISA Group
http://www.isa.us.es

In keeping with the traditional purpose of furthering science, education and research, it is the policy of the publisher, whenever possible, to permit non-commercial use and redistribution of the information contained in the documents whose copyright they own. You however are not allowed to take money for the distribution or use of these results except for a nominal charge for photocopying, sending copies, or whichever means you use redistribute them. The results in this document have been tested carefully, but they are not guaranteed for any particular purpose. The publisher or the holder of the copyright do not offer any warranties or representations, nor do they accept any liabilities with respect to them.

Support: This work was partially supported by Spanish Government under CI-CYT project Web-Factories (TIN2006-00472) and project SETI (TIN2009-07366) and by the Andalusian Government under project ISABEL (TIC-2533) and Evangelischer Entwicklungsdienst e.V. (EED).
Abstract ................................................................. vii

1 Introduction ......................................................... 1
  1.1 Software Product Lines ......................................... 1
  1.2 Variability Models ............................................... 2
    1.2.1 Feature Models ............................................. 2
    1.2.2 Orthogonal Variability Model .............................. 5
  1.3 Automated Analysis of SPL .................................... 7

2 Background .......................................................... 9
  2.1 State of the Art .................................................. 9
  2.2 Summary of Contributions ..................................... 10

3 Hypothesis and Objectives ......................................... 13
  3.1 Hypothesis ...................................................... 13
  3.2 Objectives ..................................................... 17

4 Methodology ......................................................... 19

5 Work Plan ............................................................ 21

6 Conclusions .......................................................... 23

Bibliography .......................................................... 25
List of Figures

1.1 FM example: mobile phone product line. ........................... 5
1.2 OVM example: mobile phone product line. ......................... 6
4.1 Our research methodology ....................................... 20
5.1 Work schedule summary ........................................... 21
List of Figures
List of Tables

3.1 Variability Modelling Techniques ........................................ 15
Abstract

In recent years, software engineering researchers have invested their efforts in a new line of research, known as Software Product Lines (SPL). SPL Engineering concerns a new form of software reuse which proposes a set of methods and techniques for the production of customized software products. These products are built from a common set of software artefacts in a planned and managed way. The main goal of SPL is to explore the commonalities and variabilities of a software family in order to provide strategic software reuse in such a way that improves development costs, time-to-market, and software quality.

In this new software development paradigm there is an important peculiarity, which makes it different from single software paradigm. It refers to the need for a model (a.k.a. variability model or decision model) that documents common and variable features of a SPL, and also their possible combinations. The most popular model used for this purpose is feature models (FMs), proposed in 1990. Many of the current proposals for SPL development are based on FMs. However, these models are not the only way to represent SPL variability, there are any other modelling techniques proposed in the literature, such as Orthogonal Variability Model (OVM), which documents SPL variability in an orthogonal way. Notice that OVM models don’t deal with the commonality; they only document variation points in a product line, providing an explicit view of the variations in software artefacts.

The automated analysis of SPL can be considered as the computer-aided extraction of information from variability models that can be helpful for SPL engineers, designers, programmers and managers. In spite of it is a challenge to be reached in SPL Engineering, only recently researchers have paid attention to the reasoning on these models, however their work has focused on FMs. Since OVM is another modelling language that is also being used to model SPL variability an automated support is needed to reasoning on these models as well. Although the automated analysis of OVMs has been proposed, it only deals with a small number of analysis operations, which are implemented using a specific logical representation and solver.
In the PhD thesis, our main goal is to present a proposal to achieve an adequate tool to the automated analysis on OVM. As part of it, our partial goals are, first, to formally define the OVM itself, by defining its semantics. Second, to formally define the analysis operations on OVMs, that can be found in the literature, and some additional than we have found during this dissertation. Third, we propose the extending of FAMA framework for supporting analysis on OVM. We consider that FAMA (FeAture Model Analyzer) could be a suitable option to automate this analysis since it provides a formal basis, integrate multiple solvers and already provide tools. In addition we intend to provide a support to the interoperability between OVM and FMs making use of model-to-model transformations.
Chapter 1

Introduction

In this document, we present our PhD project as a starting point for the presentation of our PhD thesis. Our work will focus on the automated analysis of Software Product Lines, particularly on Orthogonal Variability Models analysis. In addition, we will work to provide the interoperability between different kind of variability models, Feature Models and OVM. In order to place in context our work, in this Section we introduce briefly the context where it will be developed.

1.1 Software Product Lines

According to Clements and Northrop [9], a Software Product Line (SPL) is “a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way”. The main idea behind SPL is the development of a software family instead of a single software product. A SPL is composed of a set of products which are constructed from a common core asset designed for a specific domain.

In SPL Engineering the features that are common to all products, are designed as a reusable software artefact. These reusable artefacts should be flexible enough in order to add the variability required by the customer, i. e. the features that are not common to all products. Therefore, this new engineering paradigm is the latest way of software reuse, and it is rapidly gaining in importance as a viable and strategic solution for software development organizations. It promises to make improvements in time-to-market, cost, productivity and quality.
According to Clements and Northrop [9], the SPL engineering is composed of three main activities:

- **Domain Engineering**, is the activity in which the common features and the variable features are analysed and documented, and the reusable core assets are built.

- **Application Engineering**, is the activity in which the specific products are built by reusing the core assets and analysing variabilities.

- **Management**, is the activity which manages the other activities of SPL Engineering.

### 1.2 Variability Models

In SPL the variability models document the possible combinations of features in an application system. In this context, a feature might be defined as an increment in the functionality of a system. Therefore, variability models define features that represent a variation point, i.e. those features that may or may not be chosen when configuring a specific product in the application engineering activity. In addition, some variability models, such as feature models, also document features which are common to all products. Variability models are very important in SPL Engineering due to their role in documenting and managing of variability, making easier the work of management and development of a SPL.

Feature Model (FM) and Orthogonal Variability Model (OVM) are both modelling approaches employed to represent variability in SPL engineering. The former is a common approach employed to represent a SPL by means of a hierarchical decomposition of features which yields a feature tree, comprising commonalities and variabilities. The later is a more recent approach mainly used to document variability in design and realisation artefacts. Its main goal is to document variable features of the SPL without take into account the common features. In the following, we give a brief introduction to these approaches.

#### 1.2.1 Feature Models

A FM represents graphically a product line by means of combinations of features. It have been introduced by the software design community to rep-
1.2. Variability Models

resent in an abstract way the commonalities and variabilities of a SPL. A FM is composed of two main elements: features and relationships between them. Features are structured in a tree or DAG (Directed acyclic graph) where one of these features is the root. Features are connected by means of two main kinds of relationships:

- A relationship between a parent feature and a child feature.
- Cross-tree constraints, are relationships that constraint the combination of features. They use to be of kind excludes and includes. For instance, if Feature A is in the product, then the Feature B must be as well (includes), or can not be (excludes).

The first feature model was proposed in 1990 [18] as part of the method Feature-Oriented Domain Analysis (FODA). Since then, several extensions of FODA have been proposed. In FODA, a model is composed of two elements: features and relationships between them. The features are organized in a hierarchical structure in a tree. One of these features is the root and it represents the system as a whole. In this approach the relationships between features can be of two types:

i. Hierarchical relationship. This relationship is defined between a parent feature and its children. A child feature can only be part of those products in which the parent feature appears. FODA provides three types of hierarchical relationships:

   Mandatory is a relationship between a parent and a child feature which indicates that when the parent is part of a specific product, the child also must be part of it.

   Optional is a relationship between a parent and a child feature which indicates that if the parent is part of a specific product, the child may or may not be part of it.

   Alternative is a relationship between a parent and a set of child features which indicates that if the parent is part of a specific product, one and only one of the child features in the set must be in the product.

ii. Cross-tree constraint. They are of two types:

   Includes A feature X requires Y means that if the feature X is included in the product, the feature Y must be included as well, but not vice versa.

   Excludes A feature X excludes Y means that if the feature X is included in the product, the feature Y can not be included, and vice versa.
Chapter 1. Introduction

In 1998, Griss et al. [17] presented the approach Reuse-Driven Software Engineering Business, an extension of FODA adding new hierarchical relationship between parent and child features, as described below:

**Or-relationship** is a relationship between a parent and a set of child features which indicates that if the parent is part of a specific product, one or more of the child features in the set must be in the product.

Later, Kang et al. [19] proposed another extension called Feature-Oriented Reuse Method (FORM) which consider the importance of using layer in feature models to document variability in each phase of the development life cycle. To this end, they proposed the addition of four abstraction layers to the feature diagrams and new relations between features. These are described below:

**Abstraction Layers** such as *Capability layer*, *Operating Environment layer*, *Domain technology layer* or *Implementation Technique layer*. Each feature belongs to one of these four layers.

**Generalization/Specialization** are relationships that allow explicitly to define a child feature as a specialization of its parent feature, and reciprocally a parent feature to be modelled as a generalization of its child features.

**Implemented-by** is a relationship that enables features of the highest levels to connect to the features which implement them in lower levels.

In [10] and [23], the authors add to the models the multiplicity relationship to replace the alternative and or-relationships. Those relationships that represented two distinct set of features, one set called *Alternative* and another called *Or-relationship*, become only a set with multiplicity, as follows:

**Multiplicity** indicates that when the parent feature is part of a specific product, the number of child features that should be part of the product depends on the multiplicity. The multiplicity equivalent to the alternative relationship is \( (1 - 1) \), it means that one and only one of the child in the set must be part of the product. The multiplicity equivalent to the or relationship is \( (1 - n) \), where \( n \) is the number of features in the set, it means that one or more (at most \( n \)) of the features in the group must be part of the product.
1.2. Variability Models

Later, Czarnecki et al. [12, 13] proposed the cardinality-based feature models. In these works, they introduce a new hierarchical relationship which generalizes the Mandatory and the Optional relationships aforementioned.

**Cardinality relationship** indicates that when a parent feature is part of a specific product, the inclusion of its child features is dependent on the cardinality. The cardinality equivalent to the mandatory relationship is \([1..1]\), it means that the child feature must be part of the product. The cardinality equivalent to the optional relationship is \([0..1]\), it means that the child feature may or may not be part of the product. Furthermore, cardinality was added to a group of features, as was done with multiplicity, but the cardinality can have a minimum value (m) and a maximum value (n) \((\text{card}[m..n])\). For instance, if the number of child features in the group is 4, the cardinality could be \(\text{card}_4[2..3]\).

Figure §1.1 is an example of FM inspired by the mobile phone industry and graphical notation based in Czarnecki’s FM [11].

![Figure 1.1: FM example: mobile phone product line.](image)

**1.2.2 Orthogonal Variability Model**

The Orthogonal Variability Model (OVM) approach was proposed to document the variability on SPL in an orthogonal way [22]. In this model the first-classes are: *variation points* (VP) and *variants* (V). A variation point documents what vary in the SPL and a variant documents how a variation point can vary. All the variation points in the artefacts of a Product Line (PL) should be documented in a OVM model with their respective variations, but not the
commonalities. The common parts would be documented in the others arte-
fact models of the PL. Therefore, only the variability is represented by OVM
model, whilst on FMs the common parts are documented as well.

There are two abstract syntaxes defined for OVM. One of them, less for-
mal, defined by means of a metamodel [22] and the other one proposed by Metzger
et al. [21], formally defined using mathematical notation. As the metamodel
only define OVM conceptually without well formed rules, we use in this doc-
ument the Metzger’s abstract syntax to introduce OVM. Each VP must be re-
lated to at least one variant and each variant must be related to one variation
point. A mandatory VP must always be bound, i.e, all the products of the PL
must have it and its variants must always be chosen. An optional VP does not
have to be bound, it may be chosen to a specific product. Always that a VP
is bounded, its mandatory variants must be chosen and its optional variants
can, but do not have to be chosen. In OVM, optional variants may be grouped
in alternative choices. This group is associated to a cardinality $[\text{min}...\text{max}]$.
Cardinality determines how many variants may be chosen in an alternative
choice, at least $\text{min}$ and at most $\text{max}$ variants of the group. When the cardin-
ality is $[1...1]$, for default, it is not showed.

In OVM, constraints between nodes are defined graphically. A constraint
may be defined such as variants to variants, variants to variation points, and
variation points to variation points. These constraints can be excludes or re-
quires.

Figure §1.2 depicts an example of an OVM inspired by the mobile phone
industry and graphical notation based on [22].

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{ovm_example}
\caption{OVM example: mobile phone product line.}
\end{figure}
1.3 Automated Analysis of SPL

In the SPL community is well known that variability in product lines is increasing, the variability models may have thousands of features. Furthermore, these features have complex dependencies between them. Therefore, it is necessary to rely on automatic support to analyse and manage these models. The automatic analysis of SPL is the ability to reason about variability models automatically, i.e. the ability to comment on the properties of the model. For instance, we may want to know how many potential products are represented in a model, or even to know whether a specific product belongs to the model.

Automated reasoning is an important task in the context of SPL, since it is practically impossible to do it manually, and on top of that it is error prone. In addition, the variability models are one of the main artifacts of the domain engineering and therefore their analysis in an early stage of development is essential to the success of the SPL.

PhD Project Overview

This PhD project is organised as follows:

Chapter §2: Background: this chapter presents the state of the art in the context of automated analysis of software product lines. In addition, we list and comment our main contributions.

Chapter §3: Hypothesis and Objectives: in this chapter we describe our starting hypothesis to develop the thesis project. In addition, we briefly present our objectives that we want to achieve throughout the project.

Chapter §4: Methodology: in this chapter we describe the research methodology we are going to follow.

Chapter §5: Work Plan: in this chapter, we describe roughly our work plan for the thesis project, and also present a schedule summary in which are distributed the main activities to be performed.

Chapter §6: Conclusions: finally, in this chapter, we present the conclusions we have been able to draw from the preliminary research carried out.
Chapter 1. Introduction
Chapter 2

Background

2.1 State of the Art

The automated analysis of variability models is a challenge to be reached in SPLE (Software Product Line Engineering). Although there are several kinds of variability models, the majority of the research works on analysis of these models has focused on Feature Models. In the literature, there are different proposals providing automated support for the analysis of feature models [2, 4, 8, 14, 16, 20, 21, 31, 32]. In fact, around 30 analysis operations on feature models have been proposed in the recent years.

Each one of those proposals use different logical paradigm or formalism to provide the automated support (e.g. description logic, propositional logic, constraint programming). Most of them use SAT, BDD or CSP off-the-shelf solvers to automate various analysis operations, e.g. , checking if a product is valid, checking if a model is void, detecting dead features, etc.

To the best of our knowledge, there is only one proposal dealing with the automated analysis of OVM [21]. In this approach, they introduce a formalization of OVMs and propose the usage of SAT\(^1\) solvers to automate analysis operations on OVMs. This SAT solver request a Boolean formula in CNF (conjunctive normal form) and delivers all variable assignment that evaluate the input formula true. If no such assignment exists, the formula is unsatisfiable.

Furthermore, in [21] the authors provide the automated reasoning on OVM using the VFD semantics. VFD (Varied Feature Diagram) is based on FFD (Free Feature Diagrams) which is a parametric construct designed to define the syntax and semantics of FODA-inspired FD (Feature Diagram) languages

\(^{1}\)SAT4j solver http://www.sat4j.org
in a generic way [28, 29]. Metzger et al. propose reusing this formalization of feature diagrams, in other words VFD, to introduce a formalization of OVMs. They introduce a formal version of OVM's abstract syntax and describe a translation from OVM to VFD, thereby they give OVM a formal semantic. Thus, such proposal only deals with five operations, whose semantics is based on VFD feature model. They are as follows:

- **Valid model**: which checks whether a VFD is consistent, i.e. whether it allows for at least one configuration of primitive nodes.
- **Product checking**: is the verification that a given product is accepted by a VFD.
- **Commonality**: the set of primitive nodes that appear in all products.
- **Dead features**: those that do not appear in any product.
- **The number of products**: obtains all valid products conforms to the VFD.

In addition, they use just one type of solver to automate the analysis. The various solvers (SAT, BDD and CSP solvers) have varying degrees of performance and coverage with regard to analysis operations [1, 7].

As it can be seen, the support given for other kinds of variability model than feature models is lacking. There is only one proposal to analyse OVM and it addressess few analysis operations and provides a limited tool support.

To achieve a suitable automated support for the analysis of OVMs, we have identified three issues to be addressed, namely: *i)* definition of all possible operations on OVM, since only five were dealt up to now *ii)* formal representation of them, and *iii)* provide tool support to them. These aspects motivated some of the main research questions to be addressed in our Phd thesis.

### 2.2 Summary of Contributions

In the following, we comment our obtained publications for now in the context of automated analysis of Orthogonal Variability Model.

- **ASPL Workshop Paper (2008) [26]**: In this position paper, we presented a proposal that we will carry out to achieve an adequate tool to the analysis on OVMs. As part of this paper, we informally defined some analysis operations on OVMs. In addition, we proposed to study the possibility of extending FAMA framework for supporting analysis on OVMs.
2.2. Summary of Contributions

- VaMoS Workshop Paper (2009) [24]: In this paper we aimed to clarify and better explore the abstract syntax, the semantic domain and the semantic function of OVM, and to emphasize the differences between FMs and OVM concerning formal aspects.

- DSDM Workshop Paper (2009) [25]: In this paper, we proposed an algorithm to transform FM into OVM. This algorithm transforms the variable features of a FM into an OVM, thus providing an explicit view of variability of software product line. When working on these transformation, some issues came to light, such as how to preserve semantics. We discuss some of them and suggest a possible solution to transform FM into OVM by extending OVM.
Chapter 2. Background
In SPL engineering it is necessary to have clear representation for the common and variable features of the products in a product line. Variability models are predominantly used for this purpose. The Feature Models are considered one of the most important contributions to SPL modelling and the most popular variability model. However, apart from Feature Models, there are other ways to represent SPL, such as Orthogonal Variability Models.

The variability models generated in the development of a SPL can have a lot of features and a large number of combinations between them. Therefore, it is almost impossible to manage large models without an appropriate tool support. The automated analysis of SPL process consists of to extract relevant information automatically from the models, which can be useful to analysts, designers, programmers and managers of the product line.

### 3.1 Hypothesis

The automated analysis of SPLs is a key challenge in SPL engineering. In recent years there have been some proposals for this purpose, most of them focused on Feature Models. The main objective of our PhD dissertation will be giving a rigorous description of the analysis operations of variability models, particularly Orthogonal Variability Model, and extending a framework of analysis with support to such models. Based on the experience of our research group on the subject of automated analysis of SPLs using Feature Models we have raised the hypothesis to analyse SPL using other variability models. We establish a working hypothesis with a higher level of precision:

**Our hypothesis** It is possible to automate the analysis of software product
Chapter 3. Hypothesis and Objectives

lines using another variability model beyond feature models.

We intend that our PhD dissertation covers this gap and offers both a rigorous solution and a solution that supports interoperability between FMs and OVM. We want to define a set of specific operations that may be executed in the context of an automated analysis of Orthogonal Variability Models.

In order to demonstrate our hypothesis, our PhD dissertation must answer some questions. In the following we state some of them and envision their possible answers:

- **What are the variability models available in the literature?** In spite of Feature Model is the most commonly used variability technique, there are other proposals in the literature. However, it seems that little effort has been devoted to automatic analysis of these other proposals. After reviewing the state of the art we have identified a large number of techniques for variability modelling. So we have chosen five of them to study in more details, namely: OVM [22], COVAMOF [30], DecisionKing [15], K. Schmid and I. John [27], and VSL [3].

- **What differentiating criteria can be raised to choose one of the modelling techniques studied?** Based on works developed in the context of automated analysis of features models, we have chosen four criteria to decide what model we will work with in our PhD thesis. We have used five criteria, namely: *i)* if the modelling technique deals with attributes; *ii)* if the modelling technique has a formal definition; *iii)* if there are some work on automated analysis of such modelling technique; *iv)* if there is some proposal of analysis, then if it uses multiple solvers, and *v)* if the modelling technique provides some tool support. OVM is one out of the four proposals aforementioned that provide a larger number of these criteria, since it has a formal definition and provides tool support. And in addition, there is a published work about the automated analysis on OVM. However it has not been considered a wider perspective, only few operations are dealt. The result of this review is a table like the one presented in Table §3.1.

- **What analysis operations are feasible on Orthogonal Variability Modelling technique?** After reviewing the literature we have identified several operations which can be performed on OVM analysis. Some of them are: *1)* to check if a model is valid; *2)* to check if a product (set of nodes) is valid for a SPL model; *3)* to obtain the core nodes, i.e., those that are part of all products, and *4)* to check if the model has dead nodes, i.e. those which are not part of any product. In [21], these four operations
3.1. Hypothesis

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>✓</td>
<td>–</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
</tr>
<tr>
<td>Formalization</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Automated Analysis</td>
<td>✓</td>
<td>~</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Multiple Solvers</td>
<td>✓</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Tool support</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
<td>✓</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 3.1: Variability Modelling Techniques.

are formally defined. However, apart from them, several other operations applied to FMs could be performed on OVM models. We intend to identify all the analysis operations on Feature Models that can be performed on OVM models, and also to find out other possible operations on OVM.

- Which are the issues to consider when developing a suitable tool support to automate the analysis of Orthogonal Variability Models? In order to transfer our efforts to the industry we must consider some issues to implement our OVM analysis tool:

  i. Logical paradigm to support the analysis operations: currently, there are different logic paradigm or method that are already used to provide the automated support on Feature Models. In particular, we found approaches using Propositional Logic(PL), Description Logic(DL), Constraint Programming(CP), and other works based on ad-hoc solutions and algorithms. We intend to provide a formal and rigorous definition of all the analysis operations that can be performed on OVM models. Without a formal semantics, errors and misconceptions could be introduced and therefore the construction of tools is difficult and not accurate. We aim to provide
an approach that integrate more than one paradigm and/or solver because some operations may perform better with other specific algorithms or solvers.

ii. Be integrable in a tool support: we propose to extend FAMA-F framework with support for operations on the analysis of OVM. FAMA-F [4] is a formal framework for the automated analysis of software product lines in general and feature models in particular. This framework is independent of the variability model used for the analysis. FAMA-F integrates some of the most commonly used logic representations and solvers proposed in the literature (BDD\(^{1}\), SAT\(^{2}\) and CSP\(^{3}\) solvers are implemented). It integrates different solvers in order to combine the best of all of them in terms of performance. We wonder if is possible to extend this framework to automate various reasoning tasks on OVMs, e.g., verifying if a product is valid, checking if a model is void, detecting “dead” nodes, etc. We consider that FAMA-F could be a suitable option to automate the analysis of OVM since it provides a formal basis, integrate multiple solvers and has available tools.

- **Is it possible to provide a tool that allows the interoperability between both Feature Model and Orthogonal Variability Model languages?** It will be of practical importance being able to have interoperability between feature models and orthogonal variability models by means of a tool that can work with a variability model in different views. One view of all products in an SPL with their variabilities and commonalities represented with FM and another view of an SPL, where only the variation points are considered represented with OVM. In order to achieve this, we aim to use model-to-model transformation in order to generate a target model conforms to an OVM metamodel from a source model conforms to a FM metamodel. In addition, this transformation will be helpful in our research work on automated analysis of OVM. We wonder if by means of a model-to-model transformation, where from an OVM we obtain a FM, we could analyse this resulting FM by using FAMA tool. Such analysis could be thought as a subset of the automated analysis of FMs so that the analysis of a specific OVM model can yield the same results as the analysis of the equivalent FM. In spite of there is a formal definition of OVM abstract syntax and semantics, which uses mathematical notation, there is no well defined OVM metamodel. To work with model-to-model transformation we need define the metamodel of OVM in most details.

\(^{1}\)JavaBDD solver, http://javabdd.sourceforge.net
\(^{2}\)SAT4j solver http://www.sat4j.org
\(^{3}\)Constraint Satisfaction Problem www.4c.ucc.ie/
3.2 Objectives

- Elaboration of the PhD Thesis document: our main objective is the elaboration of our PhD Dissertation.

- International and National participation: Up to now we have published our first research results in some important workshops in our area, such as: Workshop on Analyses of Software Product Lines (ASPL) and International Workshop on Variability Modelling of Software-Intensive Systems (VaMoS), both in the international context, and the Taller sobre Desarrollo de Software Dirigido por Modelos (DSDM), in the national context. In the near future, we intend to participate in the main forums in our area: Software Product Lines Conference (SPLC) in the international context and JISBD as the main Spanish software engineering forum. Furthermore, we intend to present our results at the International Conference on Advanced Information Systems Engineering (CAiSE).

- International Visit: We intend to reinforce our international contacts visiting the University of Leicester by addressing shared works with Artur Boronat in the context of model transformations and software product lines. We aim to apply model transformation techniques to solve some aspects of our problem in the context of automated analysis. As a result of this visit we intend to prepare a paper to be submitted to the Software and Systems Modeling journal.

- Publications: we aim to publish our results during and after the PhD dissertation in some of the journals related with our research area, out of which we have considered 1) the Journal of Systems and Software (JSS); 2) the Software and Systems Modeling (SoSyM) journal, and 3) the Journal of the Brazilian Computer Society (JCB).

- Transference of Results: We will integrate the results of our PhD Thesis in FAMA Framework, which has been released as an open-source project and many researchers and companies have shown their interest in using it in their research projects and products.
Chapter 3. Hypothesis and Objectives
Chapter 4

Methodology

To decide about the methodology to be used in Software Engineering is not trivial because there is no consensus about the research methods to apply. Based on the experience of our research group in research and transfer of results, we have decided to combine two methodologies, the Action Research and the Unified Process. On the one hand, Action Research is an approach provided by our research group as a contribution to the debate about the problems in software engineering research [5, 6]. In this methodology two important factors to be considered are research results dissemination and technological transfer. On the other hand, the Unified Process is a framework that provides an iterative and incremental development process. It is highly appropriated for our project, since we aim for a dynamic development by applying constantly knowledge updating in a controlled way.

The first two phases of our methodology is based on Unified Process, as follows:

Inception: At this phase, we define what is the problem that we will solve. Identified the work to be done, we have to identify the key points of it. Based on our previous experience in the research context of our work, we define a possible solution. This phase was finished on the research period in which the Advanced Studies Diploma (DEA) was obtained. We carried out this activities through meetings with the PhD thesis’s supervisors.

Elaboration The main goal of this phase is to identify and define the relevant work packets and their tasks. At this phase we develop a coarse-grained project plan for the Action Research methodology. During the execution of the work project, those tasks are reviewed and refined in order to achieve the best solution.
We consider that the next two phases of the Unified Process, the Construction and Transition phases, are embedded in the Action Research methodology. At this point we use this methodology because it is focused on research results dissemination and technological transfer. Then, for each task defined in the elaboration phase we proceed with the following phases:

**Research** At this phase for each task defined in the work packets we review the literature, design a solution and validate it by means of tools, prototypes or validation tests. In addition, we determine the workshops, conferences or journals where the Research Result (RR) can be disseminated. After the validation of the RR, we disseminate it in the forums previously identified. The work method used to define steps and objectives during this phase is through periodic meetings with PhD thesis’s supervisors and research reports. In this way, the researcher will present and document the advances achieved, whether they are research results or conclusions about literature reviewed.

**Apply Research Results** At this point we establish a transfer planning of RR to the interested companies as Transfer Result (TR). We apply our main RR to real contexts, converting them into TR.

**Follow-up** At this phase, we are ready to transfer our main results to the research projects’ Observer and Promoter Companies (EPOs) in which is placed this thesis. At this point another important step is essential, to redefine our strategy based on the feedback obtained from the interested companies.

![Figure 4.1: Our research methodology.](image)
Chapter 5

Work Plan

Our work plan for the thesis period is organized to be realized in approximately two years, considering that we have already passed the lecture and research periods to obtain the Advanced Studies Diploma (DEA) in October 2008. Roughly speaking this work-plan is divided into five main groups of activities:

1) Visiting a European University
2) Prototyping the FM2OVM translator
3) Identification and formal definition of analysis operations
4) Integrating the PhD results in FAMA Framework
5) Publications
6) Versions of the thesis
7) Transfer results

Figure 5.1: Work schedule summary.

Visiting a European university: we plan to visit Artur Boronat’s group at University of Leicester in United Kingdom, from October 2009 until December 2009. This visit aims at applying model to model transformation techniques to find out a solution to our problem of interoperability between FMs and OVMs. Moreover, visiting a European University abroad, for at least three months, is one of the requirements to obtain the European PhD degree. This activity is located in the research phase of our methodology.
Prototyping the FM2OVM translator: after the research work developed during the stay in Leicester University, we intend to develop a prototype of the FM2OVM translator.

Identification and formal definition of analysis operations: in the state of the art we have identified several analysis operations which are performed on FMs. Now, during this activity, we plan to identify which analysis operations can be applied to OVM and also to identify new ones. And on top of that, formally define all those analysis operations on OVM. These work packets were identified during the elaboration phase of our methodology.

Integrating the PhD results in FAMA Framework: During this activity we plan to extend FAMA-F framework with support for operations on the analysis of OVM. Likewise, this work packet was identified during the elaboration phase and its tasks will be carried out during the research phase.

Publications: it is the activity that has more scope in the timetable. The objective of this activity is to publish all results to be obtained throughout the course of our research in the different forums presented in Section §3.2. The dissemination of our research results is an important task in our research methodology.

Writing PhD thesis dissertation: this activity will be undertaken throughout the whole schedule, however we would like to highlight two specific points in time. In mid-2010, we intend to have a preliminary version of the thesis dissertation, indicated at Preliminary version. Later in the timeline we set Final version, the point at which we believe that the final version of the thesis will be ready to submit to the doctoral committee. In order to obtain the European PhD degree, we must fulfill some requirements, such as: 1) Writing the PhD dissertation in English; 2) Send the PhD dissertation to two European doctors who will send back their insights, and 3) Proceed with the steps the University of Seville has established to inscribe an European PhD thesis.

Transfer results: We plan to present our main results to the EPOs, in two different moments: 1) when we have the first proof of concept of the FM2OVM translator, in Figure §5.1 we use TR1 to represent this moment, and 2) when we have a first proof of concept of the OVM analyser, represented by TR2.
Chapter 6
Conclusions

In this first research work we have identified that there are several proposals in the literature to represent software product lines. In addition to the known feature models, we have considered other proposals and we have found that most of them do not have a formal representation and that each one has its own concepts to model variability. The publications about these proposals have been written from different points of view, making the comparison more difficult to do. We have used some criteria to choose one of the modelling techniques, such as: having attributes, formality of the definition, providing automated analysis, and providing tool support.

In recent years the interest of the research community in the analysis of software product line is gaining importance. However, most of the works are addressed to deal with feature models. We propose provide automated analysis support to other kind of variability model, particularly orthogonal variability model. We intend to make progress in this way, by using our previous experience in automated analysis of feature models. We have identified some similarities between feature models and orthogonal variability models, thus we believe that the majority of the analysis operations on feature models can be refined to be used on OVM.

In addition, we have found that various kinds of automated support have been proposed to automate the analysis of feature models, such as different logical paradigms to implement the analysis operations and tools to automate their analysis. Regarding to the logical paradigms, we found approaches using Propositional Logic(PL), Description Logic(DL), Constraint Programming(CP), and others based on ad-hoc solutions and algorithms. After translating feature models into a logical paradigm, an off-the-shelf tool is used to automatically analyse the logical representation and subsequently the feature model. Those approaches that propose the usage of propositional logic, make use of the one of the following tools: SAT Solver, Alloy, BDD Solver, Prolog, or
SMV. Those that propose constraint programming usually make use of the one of the following tools: JaCoP Solver, Choco, OPL studio, or GNU Prolog. The proposals that use description logic paradigm found in the literature, make use of reasoners as RACER to perform their analyses.

An important motivation to this work is based on our framework FAMA-F. FAMA-F [4] is a formal framework for the automated analysis of software product lines in general and feature models in particular. This framework is independent of the variability model used for the analysis. FAMA-F integrates some of the most commonly used logic representations and solvers proposed in the literature (BDD, SAT and CSP solvers are implemented). It integrates different solvers in order to combine the best of all of them in terms of performance. Our aim extending FAMA-F with support to analysis of OVM is twofold: (i) to provide a tool support to the analysis of orthogonal variability model, and (ii) to validate our framework of analysis regarding to its abstraction, i.e. is it possible to extend FAMA-F with other variability model without change its abstract foundation layer?

In addition to the extension of the FAMA-F, we intend to work with model-to-model transformation in order to provide a tool which supports the interoperability between both languages, feature models and orthogonal variability model.
Bibliography


This document was typeset on 2009/9/2 using RC-BOOK α2.11 for LATEXe.
Should you want to use this document class, please send mail to
contact@tdg-seville.info.