On the Effort to Integrate Feature Models: An Empirical Study

Vinicius Bischoff
(PPGCA, University of Vale do Rio dos Sinos, São Leopoldo, RS, Brasil,
viniciusbischof@unisinos.br)

Kleinner Farias
(PPGCA, University of Vale do Rio dos Sinos, São Leopoldo, RS, Brasil
https://orcid.org/0000-0003-1891-3580, kleinnerfarias@unisinos.br)

Abstract: The integration of feature models plays a key role in many software development tasks, such as the evolution of software product lines by adding new features. However, little is known about the effects of the developers' experience on the integration effort and the correctness of the integration realized by them. This study, therefore, performs a controlled experiment with 25 participants (students and professionals), quantifying 250 integrations to explore two research questions, following a well-known experimental process, and quantifying the effort and correctness rate of integration of feature models realized by our participants. Our obtained results, supported by statistical tests, suggest that the number of correctly composed models and the effort invested by students and professionals are different; but this difference is not statistically significant. Our result paves the way for further studies, considering which software development tasks students and professionals obtain similar results.

Keywords: Feature models, model integration, empirical studies, controlled experiment, software modeling, integration effort

1 Introduction

The integration of software models plays an essential role in various tasks performed by software developers throughout the software product development cycle. For example, developers use model integration techniques to reconcile models that have been developed concurrently and to accommodate new features into software systems during an evolving process. In the context of collaborative software development, for example, geographically distributed teams work simultaneously on specific parts of a software design model, these parts being considered relevant for developers at any given time. The change in specific parts of the models allows developers to focus more precisely on parts of the model that need adaptations and improvements, usually required by customers to adapt the system in the face of changes in business rules.

However, at some point, the changes made in parallel will need to be integrated to generate a consolidated view of the model. Faced with the difficulty of integrating software design models, the academy has proposed some model integration techniques in the last decade, as an attempt to reduce the integration effort. Integration techniques have not been adopted by developers in practice. This means that model integration remains a manual and error-prone task. The feature model consists of a high-level model that has been widely used to represent the characteristics and their possible configurations of software products extracted from software product lines. A feature of a software system can be seen as a software functionality or expected behavior of a software system.

In this sense, the integration of feature models can be defined, broadly speaking, as a set of steps must be performed over two input models, the base model (M_A) and delta model (M_B), to produce an output intended model, M_{AB}. Figure 1 exhibits an illustrative schema of a generic model integration. The base model receives the changes contained in the delta model to become the intended model; but the base model often becomes a composed model with problems. In this sense, an extra effort effort must be spent to solve the problems, generating the intended model. Developers can integrate the input models in different ways using different model integration algorithms, or even doing this manually. Unfortunately, the output composed model (M_{CM}) and the output intended model are often different (M_{AB}). The input models tend to conflict with each other in some way due to changes done in parallel. Thus, developers need to invest some effort to detect and resolve these conflicts.

Although feature models have been used widely, little has been done to empirically analyze the effort that developers need to invest to perform integration properly. Besides, the literature does not explore the influence of the experience of developers [Filippo et al. 2010] in the effort and correctness of the integration carried out. Previous studies [Filippo et al. 2010] have investigated the influence of experience on comprehension tasks supported by UML stereotypes. Studies [Sharbaf and Zamani 2020, Mahmood et al. 2020, Abouzahra et al. 2020] indicate that model merging is still an open problem. In addition, evidence from the industry suggests that model integration and conflict resolution end up being a full-time job [Farias et al. 2015].

The software industry faces restrictions on the applicability of models in development teams [Bischoff et al. 2019, Asadi et al. 2016], which raises the following question: To what extent were the produced feature models integrated correctly? Determining the efficiency and effectiveness in the quality of the integration process, since the existing conflicts affect the comprehensibility of the models, increasing the risk of delays in software projects due to rework, as well as increasing production costs. These challenges become decisive for improving best practices in software process management.

This study, therefore, performs a controlled experiment with 25 participants, quantifying 250 integrations to explore two research questions, following a well-known experimental process, and quantifying the effort in solving problems that arise during integration and its correctness of the composed models generated by the participants. In particular, we investigate the effects of the integration of feature models, concerning the effort and correctness from the perspective of students and professionals in the context of the evolution of feature models.

The remainder of the paper is organized as follows. Section 2 outlines the main concepts discussed throughout the paper. Section 3 compares this study with others, highlighting their main differences and commonalities. Section 4 describes the adopted study methodology. Section 5 presents the study results. Section 6 introduces some implications and research opportunities drawn from our findings. Section 7 discusses some strategies followed to mitigate threats to validity of our study. Finally, Section 8 presents some conclusions and future directions.

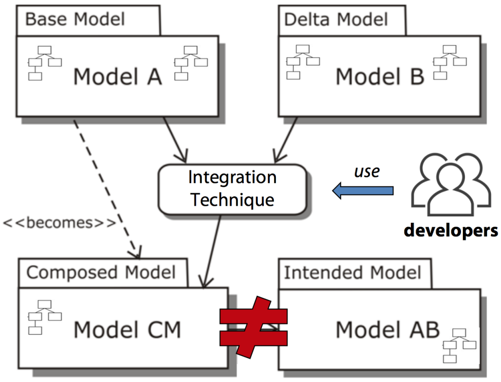


Figure 1: An illustrative schema of a generic model integration.

2 Background

2.1 Feature models

The features model is considered a high-level model used to express the products of a software product line representing the characteristics of a specific domain, its variability and similarities, as well as its relationships [Kang et al. 1990]. The main objective of the feature model is to model the common properties and possible product variables of a production line, including their interdependence [Bischoff et al. 2019, Czarnecki et al. 2002]. The features represent the attributes of the application of a given domain, being directly related and visible to the final customer [Kang et al. 1990]. A feature model can be seen as a compact representation of characteristics of a software product, and can also represent a functionality or behavioral that a software should have. The features of a software system are organized through a diagram, named feature diagram.

Moreover, the feature model resembles the structure of a tree, in which the root represents a concept and its leaves are features connected by edges that represent its state. Its status is displayed using intuitive notations to represent the points of variation. Figure 2 presents a feature model and their notations. The example illustrates the notations typically used to represent the relationships between features, mandatory, optional, exclusive and inclusive alternative, and transverse relationships, exclusion and dependency. The hierarchical relationship is defined between an ancestral feature and its descendant features. A descendant feature can only be part of a product in which its ancestral feature appears.

Figure 2 presents a model of features and their notations. The example illustrates the notations typically used to represent the relationships between features, mandatory, optional, exclusive and inclusive alternative, and transverse relationships, exclusion and dependency. The hierarchical relationship is defined between an ancestral feature and its descendant features. A descendant feature can only be part of a product in which its ancestral feature appears.

- **Mandatory**: A child feature that has a mandatory relationship, it is included in all products where the parent feature appears. In the example, the root feature *A* must create the feature *B*.

- **Optional**: The child feature can have a relationship defined as optional. Thus, it can be included optionally in all products where its main functionality is included. In the example, feature *G* may or may not be included in the software product derived from this feature model.

- **Alternative-Exclusive**: A set of child features is defined as an alternative, when only one feature can be selected, the others being excluded. The parent feature is part of the product. In the example, only feature *E* or *F* can be selected.

- **Alternative-Inclusive**: A set of child features can be added additionally to the products in which the parent feature appears. In the example, features *C* or *D* (or both) can be selected.

- **Dependency**: Selecting a feature also implies selecting another feature.

- **Exclusion**: Selecting a feature prevents you from selecting another feature.

According to the example, the root feature *A* represents a concept or functionality. The features defined below it represent the possibilities of variation existing in this domain. As can be seen, feature *B* is mandatory. Implying that it is necessary to define a feature *B*. There are features *C* and *D* below *B*. As they are inclusive alternative features, the selection of both features may occur. However, the exclusive alternative features *E* and *F*, when selected, imply the exclusion of another. As an example of an optional feature, we have feature *G*, in addition, the feature-oriented domain analysis notation [Kang et al. 1990], allows the use of dependency and exclusion between features.

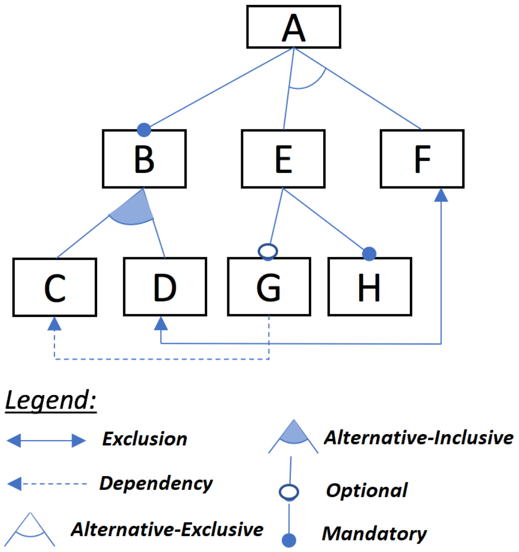


Figure 2: Example of a feature model.

2.2 Integration of feature models

The integration of feature models can be briefly defined as the set of activities that need to be performed to produce a properly integrated model [Farias 2012]. These activities are carried out on two input models (FM_A and FM_B), aiming to produce an intended or desired model (FM_{AB}) that includes requests for evolution or changes. However, the desired model is not always produced, generating an integrated model with a problem (FM_I). Efforts must be made to identify and resolve such problems to produce the desired model. Typically, the production of the intended model is not so obvious due to the presence of conflicting elements of FM_A and FM_B. We will use the terms integrated model (FM_I) and intended model (FM_{AB}) to differentiate between the output model produced with problems and the model desired. As previously mentioned, usually FM_I and FM_{AB} do not match because the input models conflict with each other in some way. The higher the number of inconsistencies in FM_I, the more distant it is from FM_{AB}. This may mean, for example, a high effort to be spent to derive FM_{AB} from FM_I (or not).

Figure 3 presents an example of integration. A developer needs to compose two feature models. The first model (FM_A) is the base feature model. The second model (FM_B) is the delta model that represents the changes that should be inserted into the base model to transform it into an output intended model (FM_{AB}). The FM_{AB} has all desired features of a particular software system. If we want to derive a software product from FM_A, this product should have the feature A and optionally the feature B. However, in the FM_B all software products to be produced must have features *A* and *C*.

Empirical studies [Farias et al. 2015, Farias et al. 2014] have revealed that the integration of design models remains a highly intensive manual task, because the model elements to-be composed usually conflict with each other in some way and such conflicts should be detected and resolved to produce an proper output model. Figure 3 illustrates two conflicts. Conflict is a contradicting value assigned to the properties of the feature models. The first conflict is that we have one feature named as B in FM_A), while we have the other feature named as *C* in FM_B. The second conflict is that the relationship between features *A* and *B* in FM_A is optional, while the relationship between the features *A* and *C* in the FM_B is mandatory.

With these conflicts at hand, software developers need to invest effort to detect and resolve these conflicts. However, usually these conflicts are not correctly understood and properly solved in real-world settings. In part, this difficulty in resolving could be explained by the lack of information about project decisions made at the time conflicts arise. Therefore, resolving conflicts becomes a challenging task.

Consequently, instead of producing an output intended model, as would be expected, the integrations end up producing an output composed model with inconsistencies. Thus, developers need to invest some extra effort to detect and resolve the created inconsistencies. In this case, the composed and intended models are inconsistent. Inconsistencies are contradicting values between the output-composed model and the output-intended model.

In this case, we have two inconsistencies: the first one would be that features B and C were inserted into the output-composed model FM_I, rather than just the feature B as would be expected in the intended model FM_{AB}. The second inconsistency is that an alternative relationship between features *A*, *B* and *C* was created, rather than a mandatory relationship between features *A* and *B*. Therefore, the output-composed model has just 25% of the output-intended model, or 75% of the output-composed model conflict with the output-intended model.

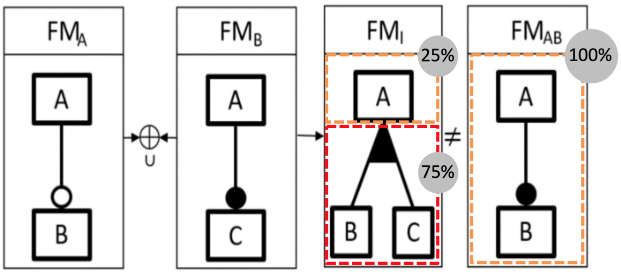


Figure 3: Example of integration of feature models.

3 Related work

This section compares the reported empirical study with the literature. For this, Section 3.1 analyses some related works. Section 3.2 introduces a comparative analysis of the related works, highlighting their commonalities and differences.

3.1 Analysis of the related works

We explored the literature to find works close to ours, considering the empirical nature and configuration of our study. In total, six articles were surveyed for convenience, using the ACM Digital Library[1], IEEE Xplore[2] and Google Scholar[3].

Farias *et al.* [Farias et al. 2015]. This study reports that one of the main limitations for the adoption of model composition techniques based on both specifications (e.g., Epsilon) and heuristics (e.g., IBM RSA) is the lack of knowledge about their effects on developers' effort. To mitigate this lack of knowledge, the article presents a controlled experiment to investigate the effort applied in different model composition techniques and detect and resolve inconsistencies in the output composed models. The techniques were evaluated with 144 evolution scenarios, producing 2,304 compositions of UML models. The findings suggest that techniques based on heuristics require less effort than techniques based on a specification to produce the intended models, in addition, there is no significant difference in the correction of composite models and the use of manual heuristics outperforms automated counterparts in the composition of models.

Farias, Garcia & Lucena [Farias et al. 2014]. This study analyses the lack of information on indicators that help developers identify models resulting from composition heuristics, with a high probability of presenting inconsistencies and understand which composed models (UML models) need more effort to be investigated through an exploratory study. This study evaluates stability as an indicator of inconsistency rate and resolution effort in model composition activities, through 180 compositions carried out to develop design models for three product lines. The results of this exploratory study indicate that stable models are a good indicator of composition inconsistency and resolution effort.

Asadi [Asadi et al. 2016]. This article studies the selection of reference models in software product lines through configuration processes. A set of visualization and

[1] https://dl.acm.org/

[2] https://ieeexplore.ieee.org/Xplore/home.jsp

[3] https://scholar.google.com/

interaction interventions is presented to represent and configure resource models, which are empirically validated to measure the impact of the proposed interventions. The empirical evaluation was carried out through a study, which follows the principles of control experiments in Software Engineering using the ISO 9126 software quality standard. The study results reveal that the visualization and interaction interventions employed improve the time to complete and change the configuration of the resource model, in addition the interventions are easy to use and learn for the study participants.

Perez *et al.* [Perez et al. 2020]. This study highlights the importance of maintenance activities embedded in Feature Location (FL) information in software artifacts. The study activities are carried out manually or automated, seeking to facilitate the maintenance tasks and evolution of typical engineering software related to features. For example, modifying and removing features in a product line, the work does not specify the integration between FL. This process consumes large amounts of time and effort, without guaranteeing good results from the development teams. The article proposes to compare manual and automated FL in a group of 18 people (5 specialists and 13 non-specialists). That is, professionals and graduate students in an industrial domain. In addition, they seek to evaluate the productivity, performance, and ease of use of both treatments in a controlled environment. The study does not evaluate the correctness rate among graduate students and professionals, as well as partially traces the implications in both treatments. Finally, the authors provide some research opportunities to improve the results of manual and automated FL techniques.

Bürdek *et al.* [Burdek et al. 2016]. This work describes the evolution of the Continuous Feature Models (FM) to meet the software requirements of the product line. The evolution of the product line leads to changes in FM. As a result, product line engineers often face problems. For example, a high rate of cohesion between the features and their semantic representation (optional, mandatory, alternative, Or), which requires great effort on the part of the team. In this context, the work presents a formal approach to compare two incoming MFs, through a case study in conjunction with experimental data. However, the work does not portray how the experiment was conducted. That is, it is understood that this is not an experiment carried out in a controlled environment. Furthermore, the authors do not present an assessment of the proposed approach between professionals (with experience) and students (without experience). Just as they do not measure inconsistencies and the effort required to use the approach. Finally, the study presents implications and future research opportunities that aim to assist the scientific community and the industry in conducting the integration and comparison of MFs.

Vyas & Sharma [Vyas and Sharma 2016]. It presents metrics to assess the usability of the Feature Models (FM), which focuses on validating a structured metric for easy and efficient use of the software product line. The metrics examined are indicators of three characteristics of usability: (1) ability to learn, (2) understand and (3) communicate. In addition to the empirical evaluation through a controlled experiment, the study involved 141 participants in an evaluation of 13 FMs. We note that all participants are students. The study did not attempt to assess the participants' efforts to understand the projected models. In addition to partially presenting implications that emerged in the course of the research point to the future, the need to conduct more experiments that lead to the comparison between the usability of MFs.

3.2 Comparative analysis of the works

This section contrasts the surveyed works with our work. This comparison, based on comparison criteria (C), serves to identify some similarities and differences. The comparison criteria are presented below:

1. **Main contribution (C1)**: Studies that have the main contribution as a case studies or controlled experiments that evaluate integration techniques and the relationship between models and their elements.
2. **Experimental study (C2)**: Studies that are a controlled experiment.
3. **Context (C3)**: Studies that were performed in a controlled environment.
4. **Participant profile (C4)**: Studies that consider students and industry professionals?
5. **Study variables (C5)**: Studies that consider effort spent and correctness of the composed models as study variables.
6. **Implications and Research Opportunities (C6)**: Does the study outline implications and point to research opportunities?
7. **Use of feature model (C7)**: Does the selected study have the features model as the target artifact of the investigation?

Table 1 presents the comparison considering these criteria. We emphasize that the the proposed empirical study was the one that most met the criteria (C1-7), highlighting its contributions and limitations.

Related Work	**Comparison Criteria**						
	C1	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**
Proposed Empirical Study	●	●	●	●	●	●	●
Farias *et al*. [Farias et al. 2015]	●	●	●	●	◐	●	○
Farias, Garcia & Lucena [Farias et al. 2014]	●	○	○	○	●	●	○
Asadi [Asadi et al. 2016]	●	●	●	○	◐	●	●
Perez *et al*. [Perez et al. 2020]	◐	◐	●	●	◐	◐	◐
Bürdek *et al*. [Burdek et al. 2016]	●	●	○	○	○	●	●
Vyas & Sharma [Vyas and Sharma 2016]	◐	●	●	◐	○	◐	●

Legend:
● Meets Fully, ○ Does not meet
◐ Meets partially, ⊘ Not Applicable

Table 1: A comparative analysis of the related works.

4 Study Methodology

This section describes the methodology used in our study. Section 4.1 describes our objective and research questions. Section 4.2 introduces the formulated hypotheses. Section 4.3 explains the study variables. Section 4.4 deals with the context and subject selection. Section 4.6 describes the analysis procedures. All these methodological steps were followed based on well-established practical guidelines about empirical studies presented in [Wohlin et al. 2012].

4.1 Objective and research questions

Our study seeks to evaluate the effects of experience on two variables: effort and correctness. These effects were investigated through evolution scenarios of feature models. In this sense, we use the GQM template [Wohlin et al. 2012] to state the objective of this evaluation, as follows:

Analyze *the experience*
for the purpose of *investigating their effects*
with respect to *effort and correctness*
from the perspective of *students and professionals*
in the context of *evolving feature models.*

Little is currently known if professionals tend to invest less effort to integrate feature models, at least while generating correctly integrated models, when compared to students, for example. Suppose a student has a success rate in an integration activity close to that of a professional. So, it makes no sense to allocate a professional if a student can achieve a similar result, requiring a lower cost. Our investigation follows this line of reasoning, seeking to understand if there is a significant difference in the results obtained by students and professionals, in terms of the effort of integration and correctness. Thus, we focus on exploring two Research Questions (RQ), as follows:

- **RQ1:** What is the effect of the experience on the integration effort?
- **RQ2:** What is the effect of the experience on the correctness of the integration?

4.2 Hypothesis formulation

To answer RQ1, we analyze one research hypothesis that investigates the impact of the experience on the effort invested to integrate feature models.

First hypothesis (H1). As mentioned earlier, the integration of feature models requires the manipulation of conflicting models, which requires certain skills to properly circumvent the situation. If conflicting changes are inadequately resolved, then models with inconsistencies are typically generated, affecting the syntactic and semantic properties in the model. A consequence of this would be the production of an integrated model that does not match the desired or intended model. Perhaps experience can be a decisive factor in circumventing conflict resolution, while it will allow you to find a coherent solution based on previous experiences. However, this is not yet evident in the context of integrating feature models. If the effort invested by more experienced people is high, then the allocation of experienced people to perform the integration of feature models becomes questionable. Perhaps the simplicity of the feature models favors people with little experience, investing an effort similar to the more experienced ones. Based on this claim, we formulate our first hypothesis.

Null Hypothesis 1, ($H1_{null}$): There is no difference in the means of the effort invested by students and professionals to integrate feature models.
$H1_{null}$: Effort(FM_A, FM_B)$_{Prof}$ = Effort(FM_A, FM_B)$_{Stud}$
Alternative Hypothesis 1, ($H1_{alt}$): There is a statistically significant difference in the means of the effort invested by students and professionals to integrate feature models.
$H1_{alt}$: Effort(FM_A, FM_B)$_{Prof}$ $\neq$ Effort(FM_A, FM_B)$_{Stud}$

Second hypothesis (H2). The second hypothesis seeks to investigate whether experience influences the integration of feature models by providing a greater number of correct integrations. In this sense, we conjecture that more experienced professionals will produce models correctly integrated in a larger number, when compared to students. This conjecture may also not be confirmed. Perhaps students perform the integration with more caution, favoring the integration of models, especially with small models. If professionals generate integrated models with a correctness rate similar to that generated by students, then it makes more sense to allocate them to more complex activities, where experience is a prerequisite.

Null Hypothesis 2, ($H2_{null}$): There is no difference in the means of the correctness rate (CorRate) produced by students and professionals when integrating feature models.
$H1_{null}$: CorRate(FM_A, FM_B $)_{Prof}$ = CorRate(FM_A, FM_B $)_{Stud}$
Alternative Hypothesis 2, ($H2_{alt}$): There is a statistically significant difference in the means of the correctness rate (CorRate) produced by students and professionals when integrating feature models.
$H1_{alt}$: CorRate(FM_A, FM_B $)_{Prof}$ $\neq$ CorRate(FM_A, FM_B $)_{Stud}$

4.3 Study variables

Table 2 presents the study variables. The independent variable of the study is the experience of the participants, which is classified as nominal, assuming two possible values: *Student* (Stud) or *Professional* (Prof). The participants were classified into two groups. Those studying in technical courses or university were considered as students. Those who exercised a professional activity were considered as professionals, highlighting that all professionals were graduated.

The dependent variables were two: effort and correction rate. The first refers to the time invested by the participants to perform integrations, assuming values from 0 to 60. Each participant had to answer 10 questions. Thus, if a participant had an effort of 15 minutes, then he needs, on average, 1.5 minutes to answer each question. The second variable quantifies the rate of the correct answer, representing the choice of a correct integration. If the participant chooses the alternative that has the integrated model correctly, then the answer is correct. The variable calculates the rate of correct answers per question. For example, if 3 out of 10 answers are correct, then the correctness rate for the question was 0.3.

Variable	Name	Scale
Independent	Main	Nominal: {Student, Professional}
Dependent	Effort	Interval [0..60]
Dependent	Correction rate	Interval [0..1]

Table 2: Study variables.

4.4 Context and subject selection

The students were also invited to participate in the experiment so that we could have subjects with different backgrounds and levels of expertise. The professionals held a Master'degree and Bachelor'degree (or equivalent) and had knowledge of software modeling and programming. The professionals were from companies located in southern Brazil, while the students from a postgraduate program in Applied Computing at the University of Vale do Rio do Sinos in Brazil. The graduate students attended one course with the following theme: Software Engineering. The experiment was part of the postgraduate course (at Unisinos) and was performed as a laboratory exercise. The authors trained participants so that everyone had a minimum level of knowledge about feature models and integration tasks.

The participants performed 10 experimental tasks related to the integration of feature models, who were familiar neither with these tasks nor with the design models. Table 2 shows the evolution scenarios describing typical tasks in which developers should evolve design models. The tasks represented cases where the participants were not the initial designers of the feature models. The models used in our study were based on different application domains, including financial and health care. Each experimental task contains an Evaluation Scenario (ES) in which two feature models (FM_A and FM_B) should be integrated. The experiment questionnaire presented 5 answer options, with the participant choosing only one option, which would represent the desired integration of the feature models.

4.5 Experimental process and design

Figure 4 shows the experimental process followed to perform the empirical study. The process is composed of three phases, activities and artifacts generated throughout the study. The subjects individually performed all activities to avoid any threat. Each activity is described as follows:

- **Training.** All participants were trained to ensure that they acquired the necessary familiarity with model integration techniques. We explain: the entire experimental process, the integration techniques, the notations used for feature models (their annotations and relationship configuration), the step by step to perform the integrations, the procedures and materials used in the experiment, including the questionnaire and how to record the time.
- **Detect Conflicts.** The second step is to analyze the FM_A and FM_B input models of each scenario based on the descriptions of changes bidding on each question, which define how the elements of FM_A were changed. Participants detect conflicts. The mediated detection effort (time in minutes) was collected during this activity, as well as a list of identified conflicts. All activity was recorded via video and audio. The records will be used to carry out the qualitative analysis.
- **Resolve Conflicts.** Participants should resolve conflicts according to the requests listed in each question to produce the intended model, FM_{AB}. The resolution effort is also measured (time in minutes) as well as video and audios are recorded.
- **Integrate Models.** This activity consists of integrating the models, producing a new model as output, FM_{AB}. The measurement of the application effort (time in minutes) is collected during this activity, stored in audio and video. After this, phase is carried out the comparative analysis between the produced model, i.e. the integrated model, FM_I, and intended model, FM_{AB} verifying if it is correct or not.

- **Interview.** Participants fill out a questionnaire, which allows them to collect information about their professional experience, academic background, modeling and development experience, gender, age, among others.
- **Material.** The models used in this experiment were feature diagrams with about 10 features, 7 relationships, 3 depth levels, and 3 conflicts, on average, by feature model. We chose to use small models for two reasons: (1) large models would require the need to control the size variable, something outside the scope of this study; and (2) controlled experiments should not use artifacts that make the activities tiring and extensive by bringing unnecessary content to the study in question.

At last, the experimental design of this study is characterized as a no repeated measure between-subjects design [Wohlin et al. 2012]. The study was organized in three steps (see Figure 4).

4.6 Analysis procedures

Quantitative analysis. After collecting the data, the first step was to make a descriptive analysis, to understand the distribution of the collected data. In this sense, descriptive statistics were produced to analyze the normal distribution [13,19]. The analysis of the normal distribution is essential when defining which statistical methods are adequate to test the formulated hypotheses. The Kolmogorov-Smirnov test was applied to pinpoint deviations from normality. We use statistical inference methods to test the formulated hypotheses. The level of significance of the hypothesis tests was $\propto = 0.05$. To test the first and second hypotheses, we applied the independent group t-test for the ten tasks. This test is similar to the Mann-Whitney, but requires two separate sets of independent and normally distributed samples. Note that we have a no repeated measure between-subjects design.

4.7 Questionnaire

The questionnaire has ten questions applied to the composition of feature models. Questions 01-06 and Question 10 present two feature models as input (MF_A and MF_B), which after completing their composition return a new model, MF_I, as Questions 07, 08 and 09 present a only feature model that supports changes in its relationship (exclusions and dependencies) between features, which seeks to know how these dependencies or exclusions affect the feature model and what the perception of analysts and developers will be. The models proposed for features occur from the derivation of a product line, which are derived from the literature (car, cell phone and the sales portal of a store). These are the models from which the participants idealized their integration, according to requirements established in each of the questions. Each question has five alternatives, and the participant can choose a single alternative, filling in the starting and ending times of each question, thus obtaining the time to perform each task.

Figure 5 presents one of the questions. We emphasize that the model of features A in relation to the model of features B, presents a semantic non-conformity, which refers to the relationship (mandatory/optional). In every question in our empirical study, the participants analyze two input feature models (MF_A and MF_B) and then choose an answer. After the execution of the tasks by the participants, the data undergoes an analysis, to calculate the integration effort and quantify the correctness rate.

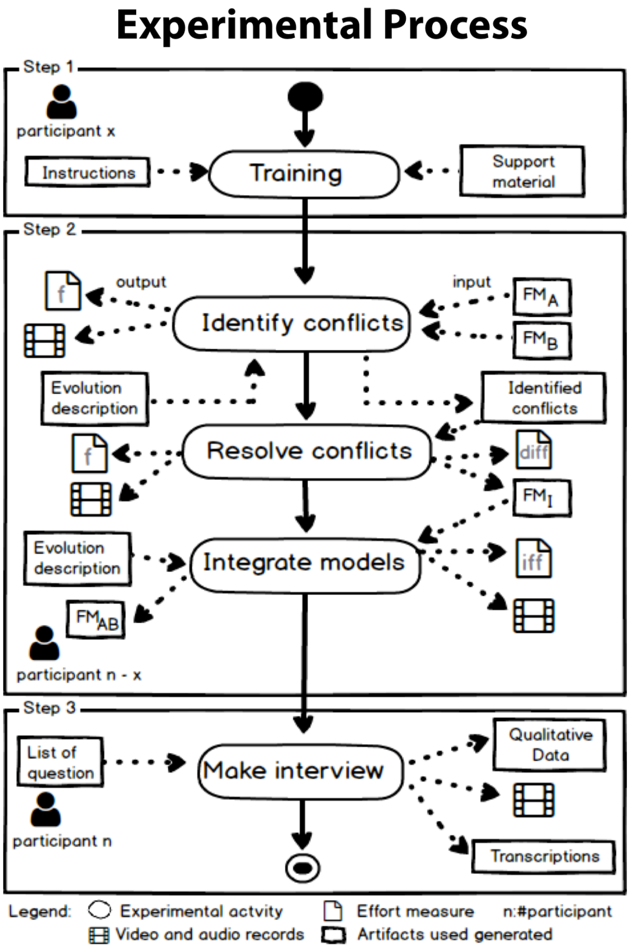


Figure 4: Experimental process followed in our study.

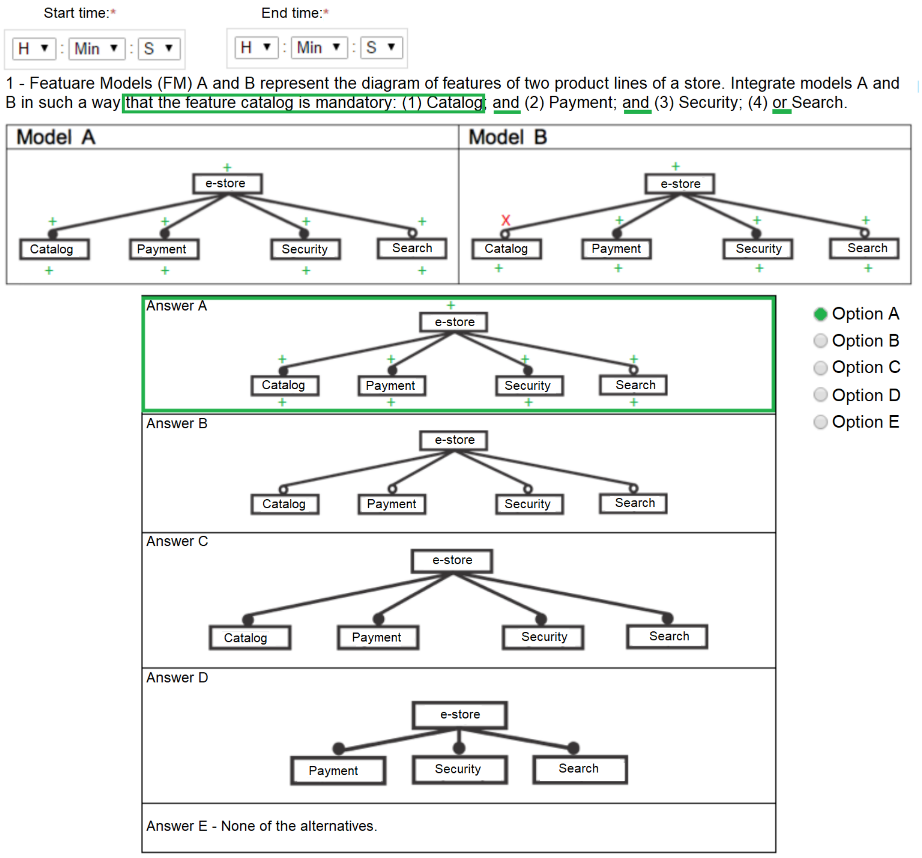
5 Study Results

This section analyzes the data set obtained from empirical study. We test the formulated hypotheses applying statistical tests using the RStudio tool[4]. Section 5.1 discusses the obtained results related to the first hypothesis. Section 5.2 presents the collected data related to the second hypothesis (H2).

5.1 Integration Effort and Experience

Descriptive statistics. This section discusses the data collected regarding the impact of the experience of the participants on the integration effort. To do this, we compute

[4] https://www.rstudio.com



Start time:* H ▾ : Min ▾ : S ▾

End time:* H ▾ : Min ▾ : S ▾

1 - Featuare Models (FM) A and B represent the diagram of features of two product lines of a store. Integrate models A and B in such a way that the feature catalog is mandatory: (1) Catalog; and (2) Payment; and (3) Security; (4) or Search.

● Option A
○ Option B
○ Option C
○ Option D
○ Option E

Answer E - None of the alternatives.

Figure 5: First question used in our empirical evaluation.

descriptive statistics to understand the distribution of the obtained data, including its main trends and dispersion. In this sense, descriptive statistics are carefully computed as grasping the data distribution and the main trends is essential. Not only the main trend was calculated using the two most used statistics to discover trends (mean and median), but also the dispersion of the data around them was also computed through the standard deviation. Table 3 exhibits the collected data related to the integration effort. Note that these statistics are calculated based on a number (N) of 10 questions, being 70 questions realized by professionals and 180 by students. The normality test of Kolmogorov-Smirnov indicates that the data are normally distributed. After analyzing these statistics, we realized that the experience had little impact on the integration effort. The main finding is that the integration effort invested by students and professionals was similar. This result is supported by some observations.

First, the mean of the integration effort invested by the students is slightly higher than the effort invested by the professionals, i.e., an increase of 14.65% comparing 2.19 (student) and 1.91 (professional). The median presented similar results. This means that students and professionals have invested a similar effort to answer the formulated

Variable	Group	N	Min	25th	Mean	Median	75th	Max	SD
Effort	Student	10	1.37	1.72	2.19	2.21	2.60	2.95	0.47
Effort	Professional	10	1	1.45	1.91	1.71	2.41	3.4	0.67

Legend:
Min: Minimum, Med: Median, Max: Maximum, SD: Standard Deviation

Table 3: The descriptive statistics of the integration effort.

questions. One implication would be that if a feature model integration activity needs to be done, then it would be indifferent in terms of effort whether it will be carried out by students or professionals. Another finding is that the effort in both cases tends to be close to the central tendency, with a standard deviation equal to 0.47 and 0.67, instead of spreading out over a large range of values. Moreover, analyzing and possibly remove outliers from the data is essential to draw out valid conclusions from the collected data. Outliers are extreme values that may influence the conclusions of our conclusions. Outliers were not found in our study.

Hypothesis testing (H1). We performed a statistical test to evaluate whether the difference between the integration effort produced by students and professionals (although small) are statistically significant. As we hypothesize that the integration effort tend to be different, the test of the mean difference will be performed as two-tailed test, considering a significance level at 0.05 ($p \leq 0.05$) to indicate a true significance. As the collected data did not violate the assumption of normality, the parametric, independent group t-test was used to test the first hypothesis. Table 4 presents the obtained results. We can see that the group means are not statistically significantly different because the value in the p-value row is higher than 0.05. Therefore, the null hypothesis ($H1_{null}$) that advocates that the effort invested by students and professionals are equal cannot be rejected. That is, there is no statistical significance to affirm that students invested more or less effort than professionals when integrating feature models.

Variable	Mean Difference	S.E. Difference	t	DF	p-value
Effort	-0.275	0.275	-1	18	0.331
Correctness	-0.132	0.098	-1.35	18	0.195

Legend:
SE: Standard Error, DF: Degree of Freedom

Table 4: The results of the hypothesis tests.

Summary for the integration effort: *The collected data indicate that the experience did not favor the reduction of the integration effort. An independent group t-test indicated that t(18) = -1, and p-value = 0.331. This implies that professionals did not invest statistically significantly lower effort (1.91 ± 0.71 min) to executing integration tasks compared to students (2.19 ± 0.49 min). The means of the two groups were not significantly different. Therefore, failing in rejecting the first null hypothesis.*

5.2 Correctness and integration techniques

Descriptive statistics. This section analyzes the obtained data regarding the impact of experience on the correctness rate. Again, we calculate descriptive statistics to reveal the data distribution, including its main trends and dispersion, as previously done. Table 5 shows the correctness of the integrations produced by students and professionals. As previously mentioned, these results are computed considering number (N) of 10 questions, being 70 questions realized by professionals and 180 by students. We performed the Kolmogorov-Smirnov test to examine the normality of the collected data. The normality test suggests that the data are normally distributed.

The main finding is that although the students had less experience, they obtained a better result. This can be explained for some reasons. First, on average the correctness rate produced by the students is slightly higher than the rate generated by the professionals, i.e., a rise of 30% comparing 0.56 (student) and 0.43 (professional). The median also favored the students, presenting an even better value than the average. Students showed an increase of 39%, comparing 0.6 (student) and 0.43 (professional). The standard deviation also showed a behavior similar to that presented in the effort variable, showing a tendency towards centralization instead of dispersion. The standard deviation is close to zero, being 0.2 (student) and 0.21 (professional). No outlier was identified in our study. Perhaps due to the simplicity of the feature model and the greater attention invested in the experiment, students were able to obtain better results. On the other hand, it may have happened that professionals, seeing the simplicity of the models, neglected the execution, not investing due attention. This result brings an interesting aspect by not following the popular wisdom that more experienced people tend to always get better results. Although the students presented a more favorable result, it is still not possible to say whether this gain without statistical significance or not. Thus, the next step was to investigate whether this result is statistically significant, thus testing our second hypothesis.

Variable	Group	N	Min	25th	Mean	Median	75th	Max	SD
Correctness	Student	10	0.23	0.6	0.56	0.60	0.70	0.9	0.2
Correctness	Professional	10	0.14	0.26	0.43	0.43	0.6	0.86	0.21

Legend:

Min: Minimum, Med: Median, Max: Maximum, SD: Standard Deviation

Table 5: The descriptive statistics of the correctness rate.

Hypothesis testing (H2). As our data is normally distributed, the independent group t-test was applied to check whether the perceived difference between the correctness rate values produced by students and professionals is statistically significant. The t-test was performed as two-tailed and with a significance level at 0.05 ($p \leq 0.05$) to indicate a true significance. Table 4 exhibits the results considering the hypothesis testing. As the p-value row is higher than 0.05, the correctness rate obtained by students and professionals are not statistically significantly different. That is, there is no significant difference between means. Therefore, our second null hypothesis ($H2_{null}$), claiming the correctness of the integrations would be equal, cannot be rejected. Our data suggests that there is no statistical significance to conclude that when integrating feature models, professionals will produce a significantly larger number of correctly integrated models.

Summary for the correctness rate: *The obtained data suggest that the level of experience did not favor the production of correctly integrated feature models. An independent group t-test indicated that t(18) = -1, and p-value = 0.195. This means that professionals did not produce a statistically significantly higher correctness rate (0.435 ± 0.225) compared to students (0.567 ± 0.212). The means of the two groups were not significantly different. Therefore, failing in rejecting the second null hypothesis.*

6 Implications and Research Opportunities

After producing and explaining the empirical knowledge produced, the next step is to draw some implications from this knowledge. For this, we will also consider our experience acquired previously through experimental studies on integration of software design models, such as the impact of aspects on inconsistency detection effort [Farias et al. 2012], the effects of model composition techniques on effort and affective states [Manica et al. 2018], the effort of composing design models of large-scale software in industrial case studies [Farias et al. 2013, Farias et al. 2012]. These implications and future directions also rely on our experience with the development of integration techniques for UML models, such as modeling language to express the merge relationship [Farias et al. 2019], an architecture for model composition techniques [Farias et al. 2018], and detection of inconsistencies in multi-view UML models [Weber et al. 2016].

We outline some research opportunities that the scientific community could explore as follows:

- **A quality model for integration of feature models.** Some quality models for design modeling have been proposed in the last decades [Lange 2007]. However, these quality models aim at software modeling in general rather than the integration of feature models itself. The further studies might extend these quality models for attending quality issues in feature modeling. This extension might be based on practical knowledge derived from developers' experience in integrating UML diagrams in practice and from researchers' knowledge in conducting empirical studies, including controlled experiments, industrial case studies, quasi-experiments, interviews, and observational studies. This evidence-based quality model might provide a guidance to developers and researchers about how to plan and run empirical studies addressing integration issues. The coming guidance might have a unifying terminology for activities and artefacts related to integration tasks, and a systematic relation between quality notions and metrics for qualitative and quantitative assessment. These terminologies and relations might help to identify and empirically evaluate possible factors or indicators of effort, accuracy, granularity and scalability of integration of feature models. For instance, a quality model might help developers to select metrics and procedures to evaluate how integration-confusing factors — i.e., the level of abstraction, domain issues, and type of techniques — would affect the precision and accuracy of the current approaches. Moreover, a quality model might also serve as a reference frame to structure empirical studies performed by other researchers in the future. Without this reference frame, the replication and contrast of empirical studies as well as the generalization of their results get impaired.

- **Insights and practical knowledge on the model comparison effort**. Based on the quality model, researchers might investigate the side-effects of integration confusing factors on the developers' effort. Some influential factors might be considered in this investigation, such as the type of integration techniques and the decomposition mechanism used to structure feature models. Moreover, researchers might also explore to what extent the rationale behind design decisions during the modeling process of feature models could influence the matching relations between the elements of feature models. Empirical findings might also enhance the knowledge about the impact of such factors on the developers'effort to apply integration techniques, detect and mitigate improper or counter-intuitive similarities between elements of feature models. A counter-intuitive similarity would be an equivalence between elements of diagrams by a comparison technique that would be contrary to developer's intuition or conventional wisdom. Additionally, we might bring together insights about how to evaluate the developers' effort, decrease error proneness realizing integration, and tame the side-effects of the influential factors in practice. The current body of knowledge on model comparison cloud be ameliorated by: (1) testing out recurring claims formulated by experts that were never evaluated; (2) identifying correlations between comparison-influencing factors and variables involved throughout the model integrations. For example, most studies to date fail to analyze which types of differences between feature models make the integration techniques more error prone by producing counter-intuitive equivalences more frequently; (3) elaborating a deep knowledge to support the formulation of theories on integration of feature models; (4) providing a solid background to inspire the creation of the next-generation integration techniques and tools; and pinpointing when the model integration techniques work and when they do not work.

- **Flexible technique to identify similarity.** Little has been done in academia to develop flexible techniques to support different strategies for identifying similarities between feature models. In addition, it is essential to develop a technique for identifying similarity between features and their relationships, which allows the application of multiple strategies of comparison, identification, and validation of similarity based on syntactic and semantic characteristics, aided by the operation of relational logic to detect and solve inconsistencies. With the application of these techniques, we will be able to seek to improve the efficiency of the algorithm, as well as perform the comparison and identification of similarity between the features and their relationships, validating the applied features model.

- **Technique for integrating feature models.** Recent literature reviews indicate that FODA notation [Bischoff et al. 2019] is the most used notation to represent feature models. However, the current literature has not given due attention to the production of effective techniques for the integration of models represented in this notation. The academic community would benefit from the development of an technique for integration of feature models. It would be very useful to have techniques that support different operations or integration methods, including joining, intersection and difference, in a semi-automatic or automatic way.

- **Easy-to-use integration tool.** Recent research [Farias et al. 2014, Farias et al. 2015] has pointed out that the current model integration tools tend to require a lot of effort from users. This turns out to be counterproductive, making the integration task costly and error-prone. Easy-to-use integration tools could reduce the effort by making the model integration process more intuitive, for example, by clearly showing similarity relationships, the well-formed rules being challenged, the overlap between the model elements, the impact of integration on quality attributes of the models, the indication of conflicts between parts of the models, and strategic information for resolving conflicting changes.

- **More empirical knowledge about integrating feature models.** Although we have presented an empirical study, further studies need to be carried out to create a plethora of evidence-based knowledge. For example, it would be interesting to conduct a controlled experiment to assess the impact of the integration technique on the effort invested by developers to produce correct models. Currently, developers can use techniques based on specification and heuristics to perform the integrations. However, little is known about the benefits of these different strategies.

7 Threats to validity

This study may have some threats to validity concerning statistical conclusion validity, construct, internal, and external threats. In this sense, we discuss some strategies used to mitigate these threats.

Statistical conclusion validity. We checked whether the independent and dependent variables were properly submitted to statistical methods. We analyzed whether the presumed cause and effect covary and how strongly they covary [Wohlin et al. 2012]. We studied the normal distribution of the collected sample, seeking to minimize the threats to the causal relation between the research variables. Thus, we verified which parametric or non-parametric statistical methods might be used. We applied the Kolmogorov–Smirnov test to check the normal distribution of the data. Since the assumptions of the statistical test were not violated, we are confident that the test statistics were chosen properly. Moreover, concerning statistical significance we tested all hypotheses considering the significance level at 0.05 level ($p \leqslant 0.05$).

Construct validity. Our main concern was on checking if we are actually measuring what we think we are measuring. By doing so, we had a certain concern about checking whether (or not) the quantification methods of the dependent variables were carefully defined, and the measures were accurately registered. The form of quantifying the dependent study variables is widely accepted in the literature, being its quantification method reused from previous work [Farias et al. 2015, Farias et al. 2019]. In addition, the experimental design used are well-documented in the literature and the experimental process is close to the previous empirical study already published [Farias et al. 2012]. Therefore, we believe that the construction of our study is reliable.

Internal validity. A causal relation involving the independent and dependent variables needs to be valid. In this sense, we sought to check that the questionnaire response preceded with the effort invested and the assertiveness of the answers, thus assuring the temporal precedence criterion. Additionally, we also observed the co-variation of the measures of the variables, i.e., the level of experience led to varying the integration effort and correctness. Still, we did not observed clear cause for the detected co-variation among study participants. Our previous experience running empirical studies [Farias 2012, Farias et al. 2013, Farias et al. 2014, Farias et al. 2015] helped us to minimize the chances of the dependent variables were affected by other existing variables, other than the level of experience. Although this is a difficult activity to guarantee, we try to do our best. In this sense, we believe that the internal validity has been carefully managed.

External validity. To what extent are the findings of this study applicable in other contexts? In this sense, the findings reported here may be considered more widely, if the context of their use is close to the configuration of the study presented in Section 4. For example, the participants need to realize integrations through questionnaires. This reality shows a not very practical perspective of our study. Despite this, our evaluated hypotheses can show that for certain activities, manipulating simple artifacts, professionals and students can obtain similar results.

8 Conclusions and Future Work

Integration of the resource model plays a key role in many software engineering activities, for example, by developing SPL design models to add new features and reconcile conflicting models developed in parallel. Many integration techniques have been proposed to support the integration of these FMs. However, we identified a lack in the literature on empirical studies on the integration of feature models. This article, therefore, reported on a controlled experiment that evaluated the effects of experience on the integration effort and the correctness of the integrations.

Our initial hypotheses were that the professionals perform the tasks with less effort and produce a higher rate of correctness than their counterpart. In total, 25 participants quantified 250 integrations to test two formulated hypotheses. Our findings indicate that the experience of students and professionals provided a difference in the effort invested and in the correct responses. Despite this, this difference was not statistically significant. Thus, we concluded that students and professionals end up having similar results when integrating simple feature models.

As future work, we intend to replicate this study with a larger number of participants. Finally, the issues outlined throughout the study can encourage other researchers to replicate our study in the future under different circumstances. We see our study as a first step in a more ambitious agenda on better supporting the integration tasks of feature models.

References

[Farias et al. 2015] Farias, K. et al.: “Evaluating the Effort of Composing Design Models: A Controlled Experiment”; Software and Systems Modeling, 2015, Vol. 14, No. 4, pp. 1349–1365.

[Sharbaf and Zamani 2020] Sharbaf, M., Zamani, B.: “Configurable three□way model merging”; 2020, Software: Practice and Experience.

[Mahmood et al. 2020] Mahmood, W. et al.: “Causes of merge conflicts: a case study of Elastic-Search”; 14th International Working Conference on Variability Modelling of Software-Intensive Systems, February, pp. 1-9, 2020.

[Abouzahra et al. 2020] Abouzahra, A., Sabraoui, A., Afdel, K.: “Model Composition in Model Driven Engineering: A systematic literature review”; Information and Software Technology, 2020.

[Kang et al. 1990] Kang, K. et al.: “Feature-oriented domain analysis (FODA) feasibility study”; No. CMU/SEI-90-TR-21. Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst, 1990.

[Czarnecki et al. 2002] Czarnecki, K., Østerbye, K., Völter, M.: “Generative programming”; European Conference on Object-Oriented Programming, 2002, June Springer, Berlin, Heidelberg, pp. 15-29.

[Bischoff et al. 2019] Bischoff, V. et al.: “Integration of feature models: A systematic mapping study”; Information and Software Technology, 2019, Vol 105, pp. 209-225.

[Farias et al. 2014] Farias, K., Garcia, A., Lucena, C.: "Effects of stability on model composition effort: an exploratory study"; Software and Systems Modeling, 2014, Vol 13, No. 4, pp. 1473-1494.

[Farias 2012] Farias, K.: "Empirical Evaluation of Effort on Composing Design Models"; PhD Thesis, 2012, Department of Informatics, PUC-Rio, Rio de Janeiro, RJ, Brazil.

[Wohlin et al. 2012] Wohlin, C. et al.: 'Experimentation in Software Engineering"; (Springer, Heidelberg, Germany, 2012).

[Farias et al. 2012] Farias, K., Garcia, A., Lucena, C.: "Evaluating the impact of aspects on inconsistency detection effort: a controlled experiment"; Proc. International Conference on Model Driven Engineering Languages and Systems, Innsbruck, Austria, September 2012, pp. 219-234.

[Filippo et al. 2010] Filippo et al.: "How developers' experience and ability influence web application comprehension tasks supported by UML stereotypes: A series of four experiments"; IEEE Transactions on Software Engineering, 2010, vol. 36, no. 1, January/February.

[Manica et al. 2018] Manica, M. et al.: "Effects of Model Composition Techniques on Effort and Affective States: A Controlled Experiment (S)"; Proc. 30th International Conference on Software Engineering and Knowledge Engineering, Redwood City, USA, January 2018, pp. 304-303.

[Farias et al. 2013] Farias, K. et al.: "Analyzing the effort of composing design models of large-scale software in industrial case studies"; Proc. International Conference on Model Driven Engineering Languages and Systems, Miami, USA, September 2013, pp. 639-655.

[Farias et al. 2012] Farias, K.: "Empirical evaluation of effort on composing design models"; Proc. ACM/IEEE 32nd International Conference on Software Engineering, Cape Town, South Africa, May 2010, Vol. 2, pp. 405-408.

[Burdek et al. 2016] Bürdek, J. et al.: "Reasoning about product-line evolution using complex feature model differences"; Automated Software Engineering, 2016, Vol. 23, pp. 687-733.

[Farias et al. 2019] Farias, K. et. al.: "UML2Merge: a UML extension for model merging"; IET Software, 2019, 13(6), pp. 575-586.

[Farias et al. 2018] Farias, K. et al.: "Toward an Architecture for Model Composition Techniques"; Proc. 27th International Conference on Software Engineering and Knowledge Engineering, Pittsburgh, USA, July 2018, pp. 656-659.

[Weber et al. 2016] Weber, V. et al.: "Detecting inconsistencies in multi-view UML models"; International Journal of Computer Science and Software Engineering, 2016, Vol. 5, No. 12, pp. 260-264.

[Lange 2007] Lange, C.: "Assessing and Improving the Quality of Modeling A Series of Empirical Studies about the UML"; PhD Thesis, Technische Universiteit Eindhoven, Eindhoven, 2007.

[Bischoff et al. 2019] Bischoff, V. et al.: "Integration of feature models: A systematic mapping study"; Information and Software Technology, 2019, Vol. 105, pp. 209-225.

[Asadi et al. 2016] Asadi, M. et al.: "The effects of visualization and interaction techniques on feature model configuration"; Empirical Software Engineering, 2016, Vol 21, pp. 1706–1743.

[Perez et al. 2020] Pérez, F. et al.: "Comparing manual and automated feature location in conceptual models: A Controlled experiment"; Information and Software Technology, 2020, Vol. 125, pp. 106-337.

[Vyas and Sharma 2016] Vyas, G., Sharma, A.: "Empirical Evaluation of Metrics to Assess Software Product Line Feature Model Usability"; International Journal of Science, Engineering and Computer Technology, 2016, Vol. 6, pp. 82.