Evaluating Variability Modeling Techniques for Dynamic Software Product Lines: A Controlled Experiment

Magno Luã de Jesus Souza*, Alcemir Rodrigues Santos*, Ivan do Carmo Machado*,
Eduardo Santana de Almeida*, Gecynalda Soares da Silva Gomes†
* Computer Science Department - Federal University of Bahia (UFBA) - Salvador, Brazil
† Statistics Department - Federal University of Bahia (UFBA) - Salvador, Brazil
Email: {magnosouza, alcemirsantos, ivanmachado, esa}@dcc.ufba.br, gecynalda@yahoo.com

***Abstract*—Dynamic Software Product Lines (DSPL) is a promising approach to enable variability management at runtime. As a particularly novel approach, variability management at runtime demands proper guidance for software engineers. Although there is a number of variability modeling techniques, understand whether they fulfill important requirements to deal with the DSPL challenges is necessary. In this work, we analyzed two variability modeling techniques with regard to their effectiveness and efficiency based on a controlled experiment conducted with 10 students. Data from performed tasks and background and feedback questionnaires were gathered and analyzed. The results showed Context-aware Feature Model technique more effective than Tropos Goal Model with Context technique considering precision. Nevertheless, both techniques were effective considering recall.**

***Index Terms*—Dynamic Software Product Lines, Dynamic Variability, Modeling Techniques, Controlled Experiment**

I. Introduction

Software Product Lines (SPL) engineering is a software development approach which provides companies with the opportunity to face economies of scale and scope, by employing a systematic reuse in the development of software products [1]. It explores commonalities shared by the software products aiming to achieve benefits such as costs reduction, improved quality and reduced time to market [2]. Moreover, SPL promotes products flexibility, which has been widely demanded by the software industry in order to produce tailor-made systems built specifically to meet particular customers needs [2]. This flexibility also known as *variability* may be defined as the ability to change or customize a system, such that it can be used in different product contexts [3].

The adaptation of the product line to generate specific products in SPL engineering is commonly handled at development time. However, emerging domains have demanded adaptations to occur during system execution, *i.e.*, at runtime. Mobile devices [4] and smart homes [5] are examples of domains that commonly demand runtime adaptations, so they can deal with either users' needs or environment constraints seamlessly [6]. Dynamic Software Product Lines (DSPL) elaborates on the SPL engineering principles by improving the capability of handling with software-variant binding at runtime [7].

As a particularly novel software development approach, DSPL engineering demands a proper guidance to support software engineers to handle dynamic adaptations. Variability modeling is an important activity of SPL engineering, which aims to represent and aid the development and reuse of variable software artifacts [8]. DSPL engineering explores the classical SPL principles and approaches [9]. However, dynamic variability management is a complex task, and introduces different challenges when compared to variability management in conventional SPL engineering. For example, the dynamic activation and deactivation of system options is a traditional way to deal with runtime variability [10], therefore, it should be captured by a particular DSPL model [10]. Besides, aspects such as context information that may help in the decision making to DSPL self-adaption should also be considered.

Although variability modeling is widely described in the SPL literature, it still lacks an adequate support for software engineers to decide the most suitable variability modeling technique for DSPL. The literature has addressed some approaches to deal with dynamic variability in DSPLs [4] [5] by adapting or extending variability modeling techniques from SPL. However, choosing the most suitable technique and understanding whether it has enough requirements to meet the needs of a given DSPL application domain is far from trivial.

In our previous work [11], we investigated the literature by identifying the main variability modeling techniques for DSPL and defined a set of evaluation criteria to rank these approaches. This rank aimed to analyze and categorize variability modeling techniques. In such an investigation, we noticed a lack of empirical studies addressing the assessment of existing techniques, particularly in the sense of techniques selection to use in complex application domains.

In this sense, this paper contributes with a controlled experiment aiming to evaluate and compare two variability modeling techniques regarding to aspects such as *effectiveness* and *efficiency*. We used measures such as *precision*, *recall* and *modeling time* to identify what most effective and efficient technique. We selected ten SPL researchers to participate of this study. Thus, they performed a task in the smart home

978-1-5090-5086-4/16 $31.00 © 2016 IEEE
DOI 10.1109/SBCARS.2016.15

domain using two different variability modeling techniques of DSPL. The data were collected through a practical task and questionnaires answers, and analyzed by means of descriptive measurements and hypotheses tests.

The results showed that *Context-aware Feature Model* technique [4] is more effective than *Tropos Goal Model with Context* technique [5] considering the precision values, whereas it was not possible to identify the most effective regarding recall, since both techniques presented similar values. On the other hand, a wide dispersion in the data set did not enable to draw conclusions regarding to efficiency for both techniques.

The paper is organized as follows. Section II presents a general background to this paper. Section III describes the experiment planning and Section IV details the operation of the study. Section V presents the data analysis by means of descriptive measurements and hypotheses testing. Section VI discusses the results and the learned lessons. Section VII discusses related work. Finally, Section VIII concludes the paper and presents some directions to future work.

II. Dynamic Variability Modeling

Variation points represent unbound options of software assets, *i.e.*, variable items that determine the software behavior through of instances in the final products [12]. Throughout the software development process, it is likely that a particular customer need or requirement triggers the need of binding a variant. The particular moment to which a variant is bound is commonly referred to as variant *binding time*. Binding times may occur before system execution, and also when the system is loaded or during its execution. In the former, the decisions in such binding times cannot be changed during system execution. Hence, they are commonly referred to as static binding times. Such a group includes pre-compilation, compilation and linking binding times. Conversely, in the latter, the decisions can be changed during system pre-execution or execution, and they are called dynamic binding times. In this group we could include, respectively, binding at load time and runtime [13].

Dynamic binding times have been the main focus of DSPL engineering. However, transitioning from static to dynamic binding times may pose several issues. The variability model is the core artifact to guide DSPL adaptations, working as the basis for generating candidate product configurations. Thus, the DSPL must be able to consult the variability model at runtime [9]. However, improving the reasoning for this model at runtime is still a challenge [14].

According to Capilla *et al.* [10], activating and deactivating system options are basic ways for a DSPL to deal with runtime adaptation. Due to these runtime changes, dependencies among system options may appear or disappear suddenly. In order to manage dynamic variability, the DSPL variability model must identify when the systems options are active, *i.e.*, their binding time. Besides, dependencies among the system options must also be considered in this model.

Other DSPL characteristics to be considered in the variability models are mutiple binding times and context-awareness [7]. The DSPL variability allows system options binding and rebinding multiple times during system execution. Context-aware and quality properties often play the role of identifying which products should vary dynamically in a DSPL. The dynamic variability modeling approaches also should provide them with an adequate support [10].

Variability modeling activity provides domain engineers with the adequate support to carry out the development of reusable software artifacts, by indicating how common and variable system options may relate with each other in the software system. It can show the path for developers to implement correct DSPL adaptations.

A. DSPL Application Domains

DSPL engineering has emerged as a promising solution to deal with a high degree of adaptability which emerging domains require. Among these domains, we could enlist mobile devices [4], smart homes [5] and web-services [15]. They should employ a self-adaptable strategy in order to deal with dynamic variations in user requirements and system environment constraints.

According to Cetina *et al.* [16], smart homes systems are highly dynamic since new types of entities such as sensors, actuators or external software systems can become necessary any time during system execution. However, these are likely to be error-prone systems and existing entities may fail for many reasons, such as hardware faults, OS errors, software bugs, and so on. The runtime variability management introduces DSPL as a solution to handle such situations in smart homes systems. DSPL engineering can use autonomic computing properties [7] in order to configure themselves automatically and detect and repair problems.

In this investigation, we analyzed the smart homes domain, due to its complexity and comprehensiveness. In practice, the smart homes is an interdisciplinary domain as it also involves domains such as Web technology, human-computer interaction, mobile computing and robotics, to ensure its operation [17]. This characteristic is particularly important to our investigation, given that we can implicitly explore a set of related domains, which may strengthen the evaluation.

The choice of a complex and widespread domain requires a robust strategy to facilitate variability management. As aforementioned, variability modeling activity plays this role. Moreover, the high degree of similarities among the different systems in smart homes domains makes variability modeling techniques a suitable strategy to deal with this issue [18].

B. Variability Modeling Techniques Selection

We earlier analyzed existing variability modeling techniques for DSPL [11], and defined some evaluation criteria to rank and characterize them. Based on such a study, we selected two techniques to use in this current work. The choice was based upon some aspects, as discussed next.

Firstly, the ranking is an important guidance, because it was conducted based on the DSPL properties listed by Hallsteinsen *et al.* [7], such as: dynamic variability; dynamic and multiple binding times; dealing with unexpected changes and

with changes by users; context awareness; autonomic or self-adaptive properties; and automatic decision-making.

Modeling smart homes projects requires a robust strategy, thus, we focused on the set of best ranked techniques. In addition, smart homes projects have shared a common property from self-adaptive systems: *context awareness* [5] [17] . Thus, among the best ranked techniques which support context information modeling, we selected the following techniques to this investigation: *Context-aware Feature Model - CFM* [4] and *Tropos Goal Model with Context - TGMC* [5].

The CFM technique extends the functional specifications of variability with contextual requirements, identified for the DSPL. It enriches the traditional *feature model* [19] with context information. In a CFM, each context information can relate with a feature through a require or exclude dependency, *i.e.*, a context can activate or deactivate a certain feature. Multiple contexts can be selected simultaneously and each combination between contexts and features represents a different configuration state of the system.

The TGMC technique analyzes variability at an early phase of software development by adopting the goal models ontology. It extends the *tropos goal model* [20] by adding context requirements, in order to capture the relationship between context and variability. The goal models are perceived as the initial sources of variability models. Tropos goal analysis designs the system as a set of actors, each one having its own strategic interests (goals). The goals are analyzed iteratively and in a top-down way to identify more specific sub-goals needed to satisfy the upper-level goals. Along this paper, CFM is called *technique A* and TGMC is called *technique B*.

In order to ensure the feasibility of using both techniques in the study, we performed the experimental activity proposed to the experiment. This preliminary investigation confirmed some aspects indicated on the ranking results [11]. Both techniques are similar with regard to dynamic elements that they can model, such as binding times and multiple binding times. The techniques also have some differences, technique B does not support to model constraints among the elements completely. Whereas technique A is among the techniques which support to model activation and deactivation of system options explicitly. Thus, both techniques were considered feasible to be used in this study.

III. Experiment Planning

Throughout this section, we describe the steps performed to plan the experimental study.

A. Goal, Questions and Metrics

This experimental study is aimed to **analyze** Variability Modeling Techniques for Dynamic Software Product Lines **for the purpose** of evaluating them **with regard to** its effectiveness and efficiency **from the viewpoint** of SPL researchers **in the context** of undergraduate, M.Sc. and Ph.D. students modeling a Smart Home DSPL project.

In order to achieve the study goal, we defined a set of research questions. Their main purpose is to characterize how the assessment should be conducted with regard to a selected quality aspect and a selected viewpoint [21]. The research questions are as follows:

RQ1. Which is the most effective DSPL variability modeling technique?
Rationale: This research question analyzes which from the two variability modeling techniques is the most effective concerning the subjects ability. We used *precision* and *recall* measures [22] to evaluate effectiveness. Precision is related to exactness of the data, whereas the recall concerns to completeness. This question is divided in two different sub-questions:

- RQ1.1. Which from the two variability modeling techniques is the most effective regarding *precision*?
- RQ1.2. Which from the two variability modeling techniques is the most effective regarding *recall*?

RQ2. Which is the most efficient DSPL variability modeling technique?
Rationale: This research question analyzes which from the two variability modeling techniques is the most efficient concerning the time each subject spent to model a DSPL. We also used precision and recall, referring to the time spent in each experimental task. The three measures are necessary in this question because, besides measuring the modeling time, it is necessary to check whether the data were modeled correctly. This question is split in two sub-questions:

- RQ2.1. Which from the two variability modeling techniques is the most efficient regarding *precision*?
- RQ2.2. Which from the two variability modeling techniques is the most efficient regarding *recall*?

In order to assess the resulting data and to answer these research questions, we defined the following metrics: M1 - *Effectiveness_PRECISION*, M2 - *Effectiveness_RECALL*, M3 - *Efficiency_PRECISION* and M4 - *Efficiency_RECALL*. These are detailed next.

M1. *Effectiveness_PRECISION* (EP). This metric aims to assess the precision of the results. We consider precision as the number of correct elements identified (true positive - TP) by the subjects over the total number of identified elements (true positives - TP and false positives - FP). Precision values range between 0% and 100%. If the precision is 100%, it means that all identified elements are correct, though there may be correct elements that were not identified [22]. This metric refers to RQ1.1 and it can be defined as:

$$EP = \frac{TP}{TP + FP} \quad (1)$$

M2. *Effectiveness_RECALL* (ER). This metrics aims to assess the recall of the results. We consider recall as the number of correct elements identified (true positives - TP) by the subjects over the total number of correct elements (true positives - TP and false negatives - FN). Recall values range between 0% and 100%. If the recall is 100%, it means that all the correct elements were identified, though there may be

TABLE I
HYPOTHESES FORMULATION

	Effectiveness Hypotheses	Efficiency Hypotheses
precision	$H1_0$: $EP_{TA} = EP_{TB}$ $H1_a$: $EP_{TA} <> EP_{TB}$ $H1_{a1}$: $EP_{TA} > EP_{TB}$ $H1_{a2}$: $EP_{TA} < EP_{TB}$	$H3_0$: $EPT_{TA} = EPT_{TB}$ $H3_a$: $EPT_{TA} <> EPT_{TB}$ $H3_{a1}$: $EPT_{TA} > EPT_{TB}$ $H3_{a2}$: $EPT_{TA} < EPT_{TB}$
recall	$H2_0$: $ER_{TA} = ER_{TB}$ $H2_a$: $ER_{TA} <> ER_{TB}$ $H2_{a1}$: $ER_{TA} > ER_{TB}$ $H2_{a1}$: $ER_{TA} < ER_{TB}$	$H4_0$: $ERT_{TA} = ERT_{TB}$ $H4_a$: $ERT_{TA} <> ERT_{TB}$ $H4_{a1}$: $ERT_{TA} > ERT_{TB}$ $H4_{a2}$: $ERT_{TA} < ERT_{TB}$

identified elements that are incorrect ones [22]. This metric refers to RQ1.2 and it can be defined as:

$$ER = \frac{TP}{TP + FN} \tag{2}$$

M3. *Efficiency_PRECISION* (EPT). This metric aims to assess the time spent (TS) for modeling based on the precision value. In this case, we needed to create a variable in order to relate the precision with the time. This relation has the purpose to validate the efficiency measure, *i.e.*, to indicate that, besides the time, the subjects have modeled either correctly or incorrectly each technique. A higher EPT implies a better modeling time regarding precision. This metric refers to RQ2.1 and it can be defined as:

$$EPT = \frac{EP}{TS} \tag{3}$$

M4. *Efficiency_RECALL* (ERT). This metric aims to assess the time spent (TS) for modeling based on the recall value. As in M3, we needed to create a variable in order to relate the recall with the time spent for modeling. A higher ERT implies a better modeling time regarding recall. This metric refers to RQ2.2 and it can be defined as:

$$ERT = \frac{ER}{TS} \tag{4}$$

In this study, the *independent variables* are the smart home DSPL project, the techniques, and the background experience of the subjects. The *dependent variables* are the number of modeled elements and the time subjects took to undertake the modeling task. The dependent variables are directly related to the measures for the hypothesis testing.

B. Hypotheses Formulation

Table I presents the hypotheses formulation. We defined a set of four different groups of null and alternative hypotheses, since each group is related to the metrics presented in the previous section. Each group has a null hypothesis where the values for both techniques are equivalent, and an alternative hypothesis where the values are different. Alternative hypotheses are divided in two others sub-hypotheses, where the first one represents a case where the value from technique A is higher than technique B and the second one represents the case where the technique B is higher than A.

C. Subjects

The subjects of this experiment were selected by convenience sampling [21]. We defined as prerequisite previous knowledge in SPL engineering. The experiment was conducted in an academic environment with students from Federal University of Bahia, as follows: one undergraduate student, three M.Sc. students and six Ph.D. students. Eight out of ten subjects hold prior background on DSPL engineering and one of them had participated in a software project using this approach. As expected, they all have been involved in SPL projects. Moreover, they hold knowledge about software modeling, since nine of them have been involved or worked on topics related to software modeling. Furthermore, eight subjects understand the role of dynamic aspects modeling, moreover, one of them was working on topics related to this approach.

D. Design

This experiment was designed by selecting one factor with two treatments. The factor is the variability modeling technique and the treatments are the techniques under evaluation. Each group was composed by 5 subjects. One group used the technique A firstly and after the technique B. Conversely, the opposite group started by using the technique B and then the technique A. The groups were randomly divided.

E. Instrumentation

Some materials[1] were used in this experiment. We used two types of forms: the *background form* to characterize the subjects according to their experience and expertise; and the *feedback form* to gather information about the techniques and the experiment execution.

In the training session, we used the *presentation content* with concepts related to the experiment. In this session, we also performed training exercises, thus, materials including a small project and instructions for their execution were given to the subjects.

In the experiment execution, we provided a *guideline* with instructions on how to model using both techniques and the experimental task description. The *task content* included the smart home project, comprising its functional and contextual requirements.

F. Experimental Activity

In this step, the subjects received a software modeling problem, based on a DSPL project in the smart homes domain. This problem was described by means of instructions for the proper system functioning. In fact, these instructions represented functional and contextual requirements, the subjects should read and interpret them. In the functional requirements, the subjects could gather functions, variability and constraints of the software system. In the contextual requirements, the subjects could gather context information and their relationship to the system functions.

[1]The experiment materials are available at http://bit.ly/DSPLmodelingstudy

Each subject performed the activity by modeling the problem using both techniques A and B. They had free time to perform it and received instructions to take notes of the start and end time for each task performed. We did not use support tools to aid this activity. The subjects received only a sheet and a pencil to perform the tasks. This activity followed the same set of the training exercise, thus, the subjects could concentrate only in the solution to the activity problem.

G. Pilot Study

The pilot study aimed to validate the experiment planning steps and to improve its execution. Moreover, the data collection method might be optimized. We used the same configuration planned for the experiment.

In this phase, we selected four different researchers to be subjects. Among them, there were one undergraduate student and three graduate students. Two subjects were more experienced than the other ones. The most experienced subjects have used DSPL approach to build software in academic projects, and a prior experience in developing SPL projects. On the other hand, the less experienced ones have never participated in a software project using neither DSPL nor SPL approaches. Nevertheless, all of them worked on topics related to software modeling as well as modeling of dynamic aspects of software systems. This difference in background helped to set the experiment, making it suitable for both most experienced and less experienced subjects.

The subjects were divided in two groups: PA and PB. Each group had a more experienced and a less experienced subject. In the group PA, the more experienced subject performed the experiment task with the technique A firstly, and the less experienced with the technique B firstly. In the group PB, the subjects did the opposite, *i.e.*, the more experienced subject used the technique B firstly, and the less experienced used the technique A firstly.

Among the lessons learned, we identified the need of more examples to ease the understanding of variability modeling techniques. Moreover, the presentation was updated and improved with a clearer and more understandable content. Based on the execution of the experimental task by the subjects, we reduced the number of activities required in the study, because the subjects took a longer time than expected to carry out the tasks. In fact, we reduced the number of functional and contextual requirements existing in the task. According to the data gathered from the pilot study, the reduction in the task elements did not cause negative impacts in the study evaluation.

The improvements in the data collection method was one of the most important contributions from the pilot study. After the execution of this pilot, we analyzed the data, and realized the necessity to create four different categories. It helped to better organize the data and promote a better comparison among both techniques.

IV. Experiment Operation

Throughout this section, we describe the steps performed in the experiment operation. Thus, the tasks performed by subjects and the data collection process are described.

TABLE II
Experimental study agenda

	Topic	Description	Length of time (hour:minute)
A	Consent	- Introduction - Consent	0:10
B	Training	- Concepts - Exercises	2:30
C	Background	- Background Form	0:15
D	Experimental Activity	- Execution of the activity.	Free
E	Feedback	- Feedback Form	0:15

A. Execution

Table II presents the experimental study agenda. At the beginning, we explained the main objectives of the experiment (A), such as the topics the study would address and the structure of the experiment execution. Detailed information such as goal, research questions, and hypotheses were not mentioned in order to reduce any potential bias in the study. The subjects signed a consent form. It took us 10 minutes to accomplish this initial task.

Next, we conducted the training phase (B), comprising concepts about DSPL and variability modeling techniques. We also applied exercises in a similar domain in order to promote a better understanding of how to model variability using both approaches. This phase took 2:30h to accomplish.

After, we applied the background form (C) aiming to characterize the subjects profile. We collected data about their experience in SPL, DSPL, software modeling and dynamic aspect modeling. This phase took 15 minutes to accomplish.

Finally, we applied the experimental task (D). The subjects were divided in two groups. Each group received a document with instructions about the activity, including the functional and contextual requirements for the modeling task. Besides, the subjects had access to a guideline for modeling using both techniques and one researcher was available to answer any questions and solve doubts about the activity. The time was free in this phase, since we also collected data related to the time spent to model the activities.

Next, we applied the feedback form (E), where the subjects reported strengths and weaknesses of each variability modeling technique, and difficulties and problems found in the experiment execution. This phase took 15 minutes to accomplish.

B. Data Collection

We performed the data collection based on the subjects answers from background and feedback forms and from the experimental task answers. We divided the data collection in four different categories: *functional requirements*, which identifies functions of the systems represented by features in the technique A and by goals and tasks in technique B; *contextual requirements*, which represents the contextual requirements of the system; *relationship*, which represents the

TABLE III
RAW RESULTS FOR EACH SUBJECT

SB	TC	TS	GP	PC	RC	PC/TS	RC/TS
01	A	26	G1	88%	87%	0,034	0,033
01	B	31	G1	65%	86%	0,021	0,028
02	A	42	G1	97%	100%	0,023	0,024
02	B	19	G1	83%	93%	0,044	0,049
03	A	39	G1	92%	97%	0,024	0,025
03	B	25	G1	76%	98%	0,030	0,039
04	A	20	G1	95%	86%	0,048	0,043
04	B	17	G1	63%	67%	0,037	0,039
05	A	31	G1	95%	95%	0,031	0,031
05	B	25	G1	70%	87%	0,028	0,035
06	A	24	G2	90%	93%	0,038	0,039
06	B	22	G2	86%	98%	0,039	0,045
07	A	21	G2	69%	80%	0,033	0,038
07	B	21	G2	73%	79%	0,035	0,038
08	A	22	G2	79%	88%	0,036	0,040
08	B	22	G2	58%	97%	0,026	0,044
09	A	26	G2	94%	95%	0,036	0,037
09	B	35	G2	80%	98%	0,023	0,028
10	A	23	G2	85%	80%	0,037	0,035
10	B	56	G2	73%	93%	0,013	0,017

Subtitle: SB - Subject; TC - Technique; TS - Time Spent; GP - Group; PC - Precision; RC - Recall.

relationship among the different elements from each technique; and *variability*, related to how the variability in both techniques are expressed.

Thus, in order to collect the data according to this division and calculate the precision and recall, we identified three different values in each category: *True Positive* (TP), *False Positive* (FP), and *False Negative* (FN).

V. DATA ANALYSIS

Along this Section, we present the data analysis. Table III presents raw data gathered from each subject. For the sake of comprehension, we then labeled subjects 01 to 05 for the set of participants from the first group (G1) , and subjects 06 to 10 for the participants from the second one (G2). The subjects reported their time spent in each task by taking notes of the start and end time. We then calculated the measures EP, ER, EPT, ERT. In the next subsections, we elaborate on these measures, by describing their gathered values and the descriptive statistics, as well as a discussion surrounding the hypotheses tests.

A. Effectiveness Analysis

In order to answer the first research question, we used EP and ER measures. Table IV presents the results of the descriptive measurements for *mean*, *standard deviation* and *coefficient of variation* values.

The **precision** results show that technique A yielded 88.4 as a mean value, while technique B yielded 72.7 as a mean value, this means the subjects were able to precisely model 88.4% of the elements using technique A and only 72.7% of the elements using technique B. In addition, the coefficient of variation, which is a standardized measurement of dispersion and takes in consideration both mean and standard variation in its calculation, has presented significantly low values for both techniques. However, the technique A presented a coefficient of variation of 9.9% whereas the technique B presented 12.3%. Thus, it indicates that technique A has a more homogeneous data set than the latter.

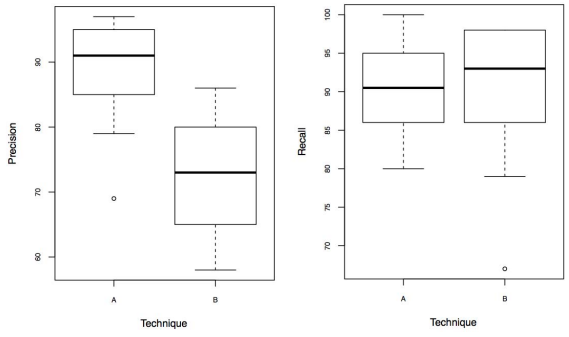
TABLE IV
DESCRIPTIVE MEASUREMENTS

Variables	Techniques	Mean	SD	CV (%)
Precision	A	88.40	8.72	9.9
	B	72.70	8.97	12.3
Recall	A	90.10	6.97	7.7
	B	89.60	10.18	11.4

Subtitle: SD - Standard Deviation; CV - Coefficient of Variation.

The **recall** results show that technique A and technique B are similar in terms of *mean*, technique A obtained 90.1 which is slightly higher than 89.6 from technique B. It means that the subjects were able to model 90.1% of the elements using technique A, whereas they modeled 89.6% of the elements using technique B based on recall values. Furthermore, the coefficient of variation related to recall presented considerably low values for both techniques, but technique A obtained a lower value, 7.7%, while technique B got 11.4%, which means that the technique A is a bit more homogeneous than technique B based on recall.

Figure 1 shows boxplots built from recall and precision raw data. Figure 1(a) shows the precision boxplot graphic for both techniques. The larger part of the data from technique A is above a larger part of the data from technique B. It reinforces the analysis of precision according to the mean values. Besides, the length of the technique B box is bigger than the technique A box, meaning a higher dispersion of the data in technique B. The boxplot of technique A is skewed, the lower whisker is longer than the upper one and the longer part of the box is below the median line, meaning that most of the data in technique A are below of its median. On the other hand, the boxplot of technique B shows that its set of data are symmetric, since the median line cuts the box in the middle.



(a) precision (b) recall

Fig. 1. Boxplots presenting the EP and ER values

TABLE V
SHAPIRO-WILK TEST FOR PRECISION AND RECALL VARIABLES

Variables	Shapiro-Wilk normality	p-value
Precision	0.9522	0.4015
Recall	0.8940	0.0319

Subject 07 was an outlier in the use of technique A. The raw data investigation gave us no reason to delete this subject from the analysis. The subject considered the technique A robust and easy to use, although it declared himself undecided when asked about the ease of learning. Moreover, he/she obtained a good performance in technique B with a precision mean of 73% (Table III) which is equivalent to median and above the mean of this approach.

Figure 1(b) shows the recall boxplot for both techniques. The minimal and maximum values from technique A are higher than from technique B, although the median line of this technique is slightly above the other technique. Besides, it is difficult to make inferences about the median value between technique A and technique B, as well as the mean value, the difference is small. Technique B results are more spread than technique A results, if we consider the outlier. Technique A box shows that its data set is symmetric, whereas technique B box is skewed, the lower whisker is longer than the upper one and the longer part of the box is below of the median line, meaning that most of the data in technique B are below of its median.

Subject 04 was an outlier in the use of technique B. The raw data investigation gave us no reason to delete this subject from the analysis. This subject disagreed when asked about technique B robustness in the feedback form, and it declared himself undecided about ease of use and ease of learning for this approach. Moreover, it considered the way of modeling of this technique confusing and inaccurate, and it reported the necessity of addressing the exclude relationship which is available in feature models. Besides, he/she obtained a good performance in the modeling task using technique A with a recall mean of 86%.

B. Effectiveness Hypothesis Testing

We performed the hypothesis testing to verify if there is a significant difference between the data set of both techniques regarding effectiveness [21]. Table V shows the results from Shapiro-Wilk test which evaluates the variables normality. The normality of the data was evaluated aiming to decide which the most suitable type of testing. The testing indicates that the *precision* variable is no significant to the level of 5% (p-value < 0.05), *i.e.*, its data set has a normal distribution.

In order to test the data with a non-normal distribution, we used the non-parametric Mann-Whitney U test. To test data with a normal distribution, we used the Student's t-test [21]. Table VI shows the testing results, since for the *precision* variable were used both tests. Only the *precision* variable presented a significant difference to the level of 5% for both

TABLE VI
HYPOTHESES TESTING FOR PRECISION AND RECALL VALUES

Variables	t-test	p-value	Mann-Whitney U	p-value
Precision	3.9686	0.0009	89.0	0.0036
Recall	-	-	47.5	0.8792

TABLE VII
DESCRIPTIVE MEASUREMENTS

Variables	Techniques	Mean	SD	CV%
Precision/Time	A	0.03381	0.0071	21.0
	B	0.02962	0.0082	27.7
Recall/Time	A	0.03440	0.0063	18.3
	B	0.03610	0.0067	18.6

Subtitle: SD - Standard Deviation; CV - Coefficient of Variation.

Mann-Whitney U and Student's t-test. It means we could only refute the precision null hypothesis.

C. Efficiency Analysis

In order to answer the second research question, we used the ratio value between precision and time spent and between recall and time spent. Table VII shows the results of the descriptive measurements with the *mean*, *standard deviation* and *coefficient of variation* values.

The **precision over time** results show that technique A obtained a mean of 0.03381 while the technique B obtained a mean of 0.02962, meaning that the users were more efficient in performing the task using technique A considering precision over time. Besides, the coefficient of variation presented smaller values for technique A (21%) than technique B with 27.7%, meaning that technique A had a more homogeneous data according to this analysis.

According to **recall over time**, the results show that both techniques are similar considering the mean value. Technique A obtained a mean value of 0.03440, which is slightly smaller than technique B with a mean of 0.03610, meaning that the subjects were most efficient for modeling the task elements using technique B considering the recall. The coefficient of variation for both techniques were quite similar ones: technique A obtained 18.3%, while technique B obtained 18.6%, meaning that technique A had data slightly more homogeneous than technique B according to this analysis.

Figure 2 shows the boxplots for precision and recall over time. Figure 2(a) shows the precision over time boxplot graphic related to results from both techniques. The length of technique B box is slightly bigger than technique A box, meaning that the set of data for technique B varies more than technique A. The boxplot of technique A is skewed, the lower whisker is longer than the upper one and the longer part of the box is below of the median line, meaning that most of the data in technique A are below of its median value. In technique B box, both lower whisker and upper whisker have similar length, but the longer part of the box is slightly above the median line. Thus, most of data in technique B is above of its median. Moreover, technique A median is above

TABLE VIII
SHAPIRO-WILK TEST FOR PRECISION OVER TIME AND RECALL OVER TIME

Variables	Shapiro-Wilk normality	p-value
Precision/Time	0.9767	0.8847
Recall/Time	0.9674	0.6984

TABLE IX
HYPOTHESES TESTING FOR PRECISION AND RECALL OVER TIME VALUES

Variables	t-test	p-value	Mann-Whitney U	p-value
Precision	1.1317	0.2737	63.0	0.3527
Recall	-0.4657	0.6479	39.0	0.4359

technique B median. The graphic shows an outlier (Subject 04) for technique A box. We investigated the experiment data and we did not find any other reason than a high performance of this subject.

Figure 2(b) shows the recall over time boxplot graphic related to results from both techniques. The length of the technique B box is bigger than technique A one, thus, the range of data of technique B varies more than technique A. However, the median line of technique B is above the median line of technique A. Both techniques boxes are skewed, their lower whisker is longer than the upper one and the longer part of the box is below of the median line. It is more evident in the technique B box, meaning that most of the data in both techniques are below of their median value.

D. Efficiency Hypothesis Testing

We performed the hypothesis testing to verify if there is a significant difference between the data set of both techniques regarding to efficiency [21]. Table VIII shows the results from Shapiro-Wilk normality test. The testing indicated that both variables are not significant to the level of 5%, i.e., both data set have normal distribution.

In this case, we can use both Mann-Whitney U and Student's t-tests. Table IX shows the testing results. Both *precision over time* and *recall over time* variables did not present a significant difference for both Mann-Whitney U and Student's t-test to refute the null hypothesis.

VI. DISCUSSION AND LESSONS LEARNED

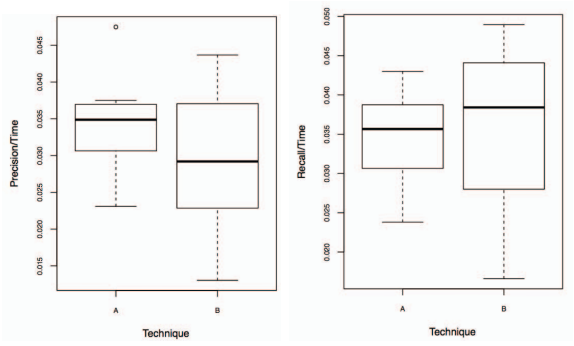
The results show technique A more **effective** than technique B considering the yielded *precision* values. However, if we consider *recall* variable, gathered evidence may not be clear enough to make us draw significant conclusions on this matter. Although data from technique A were slightly more homogeneous than data from technique B, the recall mean values from these techniques are similar. In addition, the hypothesis testing for the recall variable confirms it, since it was not possible to refute the null hypothesis. The opposite happened with the precision variable, the null hypothesis was refuted in the tests.

The results indicate that the subjects were capable to model most of elements of the DSPL project using both techniques in this study. However, the subjects by using technique A were capable to represent them more precisely.

Regarding **efficiency**, technique A took advantage when compared to the technique B in the *precision over time* variable. However, both techniques have data sets widely disperse considering the coefficient of variation. Regarding *recall over time* variable, both techniques present similar values for mean, and their data sets are too similar and disperse when the coefficient of variation is considered. Thus, the *precision over time* and *recall over time* variables did not present significant difference between data sets to refute the null hypotheses in the hypothesis test. The results hinder any conclusions about the efficiency of both techniques.

A. Subjects Feedback

The feedback form was composed by closed and open questions. Closed questions were divided in multiple and agree-disagree choices ranging from totally disagree to totally agree. Figure 3 presents the feedback form answers. When questioned about the **robustness**, all the subjects reported that they believe that the technique A is robust enough for modeling DSPL, whereas only 30% of the subjects considered



(a) precision over time (b) recall over time

Fig. 2. Boxplots presenting the EPT and ERT values

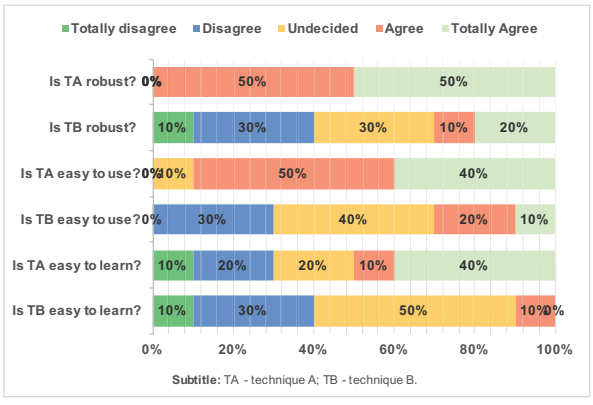


Fig. 3. Closed questions from the feedback form

the technique B robust enough for it. In addition, in the multiple choice question, 90% of the subjects reported that consider technique A more robust than technique B.

Concerning to the **easiness of use**, 90% of the subjects consider the technique A easy to use, while only 30% agreed that the technique B is easy to use in the DSPL modeling task. When the approaches were compared, 100% of the subjects reported that consider the technique A the easiest to use. Regarding **easiness of learning**, 50% of the subjects considered the technique A easy to learn, whereas only 10% answered it for technique B. Again, all the subjects chose technique A to technique B when asked about which one is more easy to learn.

The open questions revealed that 30% of the subjects complained about the relationship among the features or between the features and contexts in the technique A, and 20% mentioned about the need of a tool support for this approach. 40% of the subjects suggested a way for modeling constraints among the system elements in technique B, 50% complained about some difficulties for modeling context information, 10% reported difficulty for modeling the relationship between goals and tasks, and 10% suggested a tool to support this approach. When asked about what is the most efficient technique, 90% of the subjects indicated technique A. The only subject that considered technique B more efficient believes it is most directly connected to software implementation.

Thus, most of the subjects agreed technique A is more robust, easier to use and to learn than technique B. However, according to some subjects, technique A needs a better way to deal with relationships between features and contexts. They also pointed out a need of tool support. We believe these favorable answers to technique A is mainly due to two reasons: firstly, technique A was capable to model a DSPL project more precisely than technique B and these answers confirm its effectiveness; next, technique A is an extension of feature model that is traditionally used by the SPL research community.

B. Threats to validity

We next discuss the threats to the validity of this empirical evaluation.

1) Internal Validity: An important threat considered in this study was the maturation. As the sample for this study was small, we decided that all subjects should perform both treatments. Aiming to mitigate the learning effect, we divided the subjects in two groups where each one applied the treatments in a different order, thus, the learning effect was balanced.

2) External Validity: As this empirical evaluation was carried out in an academic environment, and with constrained settings, it might be difficult to generalize its findings. However, as one of the few studies in the field, the results may be used as baselines for further replications. Besides, the experimental package may provide researchers with an adequate support in further replications of this study.

3) Construct Validity: The experimenter expectancies may be considered as an important threat affecting the construct validity, as it might bias the results. In order to mitigate such a threat, we decided to involve people with a strong interest in the topic under evaluation. The choice of the two techniques is another potential threat. We cannot ensure those are the most representative ones of the field. However, the techniques selection is backed up with empirics from a previous investigation [11].

4) Conclusion Validity: The random heterogeneity of the subjects is a potential threat affecting the conclusion validity. In this sense, we selected subjects with a similar background and applied a training session as an attempt to balance their knowledge. Nevertheless, as most of the subjects hold a certain knowledge in SPL engineering, such a background might have influenced the results, given that the technique A extends the widely-accepted Feature Modeling approach [19]. In this effect, further empirical studies are encouraged, in which subjects without such a background should be involved.

VII. Related Work

To the best of our knowledge, this work is the first empirical study aiming to evaluate DSPL variability modeling techniques. Nevertheless, Hadar *et al.* [23] evaluated the comprehensibility of requirements models by comparing two different modeling approaches: *Use Case* scenario-based and *Tropos* goal-oriented modeling. Other aspects such as effort required to comprehend both approaches and the derived productivity in each case also were evaluated. Similarly to our work, measures such as precision, recall and modeling time were used in the evaluation. This study was performed by means of a family of controlled experiments with a first experiment and two replications.

Sinnema and Deelstra *et al.* [8] proposed a framework for classifying six variability modeling techniques regarding their common and different characteristics. This framework encompassed only concepts related to static variability modeling. Alves *et al.* [14] analyzed commonality of variability management between SPL and Runtime Adaptable Systems. They discussed on the feasibility of integrating some aspects of both approaches, such as: a more systematic approach towards variable binding time; and the formalization of context information and its relation to product variants. Although these work discussed variability modeling approaches, and even in some cases addressed dynamic aspects of variability, they did not focus on DSPL approach.

Bencomo *et al.* [6] analyzed three studies [14] [24] [25] conducted with distinct research methods to characterize the maturity of the field and identify main contributions and gaps. This work provided an important overview about the area of DSPL addressing issues such as the feasibility of achieving runtime variability with current DSPL-oriented approaches. Capilla *et al.* [10] provided an overview of the state of the art and current techniques proposed to deal with the several challenges of runtime variability mechanisms on the DSPL context. Besides the challenges, possible solutions to support runtime variability mechanisms in DSPL models were discussed. Although DSPL variability modeling was not the

main focus of these work, it was addressed and discussed aiming to achieve solutions and improvements to the activity.

VIII. Conclusions and Future Work

In [11] we presented an analysis and characterization of variability modeling techniques to DSPL using a ranking approach. We selected the main variability modeling techniques from the literature. Next, we used a set of criteria identified according to the main properties of DSPL. Based on this ranking, we picked *Context-aware Feature Model - CFM* [4] and *Tropos Goal Model with Context - TGMC* [5] techniques out to use in this experimental study. An initial investigation on variability modeling techniques for DSPL was provided by carrying out a controlled experiment by evaluating their effectiveness and efficiency.

The results of this study indicated that subjects were more effective by using CFM technique than TGMC technique considering precision, *i.e.*, through the technique A is possible to model more precisely the elements of DSPL. Regarding recall, both techniques obtained positive results, meaning that they are capable to model most of elements of DSPL. However, it was not possible to identify what is the most efficient technique. It occurred due the dispersion of data set in both variables precision and recall over time.

Our controlled experiment revealed some research opportunities, which can be explored in future work. It is necessary to replicate with a larger sample of subjects. In addition, replications in the industrial scenario with software engineering professionals should be considered. Thus, we could leverage stronger evidence concerning the techniques' efficiency. We also plan to address new variability modeling techniques as well as an additional experimental object, *i.e.*, a different application domain in next replications in order to enrich the work results. On the basis of subjects' feedbacks, we could also carry out a qualitative evaluation.

We believe that the knowledge obtained with these empirical studies could contribute with improvements in the existing variability modeling techniques. Moreover, the development of variability modeling guidelines could help researchers and practitioners of DSPL modeling to conduct their work. This can contribute with the maturity in the field of dynamic variability modeling and benefit DSPL and others areas which have required to deal with variability dynamic aspects.

Acknowledgments

This work was partially supported by the National Institute of Science and Technology for Software Engineering (INES[2]), funded by CNPq and FAPESB.

References

[1] P. Clements and J. McGregor, "Better, faster, cheaper: Pick any three," *Business Horizons*, vol. 55, no. 2, pp. 201 – 208, 2012.

[2] P. Clements and L. Northrop, *Software Product Lines: Practices and Patterns*. Boston, MA, USA: Addison-Wesley, 2001.

[3] J. Van Gurp, J. Bosch, and M. Svahnberg, "On the notion of variability in software product lines," in *Working IEEE/IFIP Conference on Software Architecture.*, 2001, pp. 45–54.

[4] K. Saller, M. Lochau, and I. Reimund, "Context-aware dspls: Model-based runtime adaptation for resource-constrained systems," in *17th International Software Product Line Conference*, ser. SPLC '13 Workshops. ACM, 2013, pp. 106–113.

[5] R. Ali, R. Chitchyan, and P. Giorgini, "Context for goal-level product line derivation," in *3rd Workshop on Dynamic Software Product Lines*, 2009.

[6] N. Bencomo, S. Hallsteinsen, and E. Santana de Almeida, "A view of the dynamic software product line landscape," *Computer*, vol. 45, no. 10, pp. 36–41, 2012.

[7] S. Hallsteinsen, M. Hinchey, S. Park, and K. Schmid, "Dynamic software product lines," *Computer*, vol. 41, no. 4, pp. 93–95, 2008.

[8] M. Sinnema and S. Deelstra, "Classifying variability modeling techniques," *Information and Software Technology*, vol. 49, no. 7, pp. 717–739, 2007.

[9] M. Hinchey, S. Park, and K. Schmid, "Building dynamic software product lines," *Computer*, vol. 45, no. 10, pp. 22–26, 2012.

[10] R. Capilla, J. Bosch, P. Trinidad, A. Ruiz-Cortés, and M. Hinchey, "An overview of dynamic software product line architectures and techniques: Observations from research and industry," *Journal of Systems and Software*, vol. 91, pp. 3–23, 2014.

[11] M. L. J. Souza, A. R. Santos, and E. S. Almeida, "Towards the selection of modeling techniques for dynamic software product lines," in *Proceedings of the 5th International Workshop on Product LinE Approaches in Software Engineering*. IEEE Press, 2015, pp. 19–22.

[12] C. W. Krueger, *Product Line Binding Times: What You Don't Know Can Hurt You*, 2004, pp. 305–306.

[13] K. Czarnecki, P. Grünbacher, R. Rabiser, K. Schmid, and A. Wasowski, "Cool features and tough decisions: A comparison of variability modeling approaches," in *Workshop on Variability Modeling of Software-Intensive Systems*, 2012, pp. 173–182.

[14] V. Alves, D. Schneider, M. Becker, N. Bencomo, and P. Grace, "Comparitive study of variability management in software product lines and runtime adaptable systems." in *Workshop on Variability Modelling of Software-intensive Systems (VaMoS)*, vol. 29, 2009, pp. 9–17.

[15] G. Alferez and V. Pelechano, "Context-aware autonomous web services in software product lines," in *15th International Software Product Line Conference (SPLC)*, 2011, pp. 100–109.

[16] C. Cetina, J. Fons, and V. Pelechano, "Applying software product lines to build autonomic pervasive systems," in *12th International Software Product Line Conference*, 2008, pp. 117–126.

[17] M. Chan, D. Estève, C. Escriba, and E. Campo, "A review of smart homespresent state and future challenges," *Computer methods and programs in biomedicine*, vol. 91, no. 1, pp. 55–81, 2008.

[18] C. Cetina, P. Giner, J. Fons, and V. Pelechano, "Autonomic computing through reuse of variability models at runtime: The case of smart homes," *Computer*, vol. 42, no. 10, pp. 37–43, 2009.

[19] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson, "Feature-oriented domain analysis (FODA) feasibility study," Carnegie-Mellon University Software Engineering Institute, Tech. Rep., 1990.

[20] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, "Tropos: An agent-oriented software development methodology," *Autonomous Agents and Multi-Agent Systems*, vol. 8, no. 3, pp. 203–236, 2004.

[21] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*. Springer, 2012.

[22] A. De Lucia, F. Fasano, G. Scanniello, and G. Tortora, "Comparing inspection methods using controlled experiments," in *12th International Evaluation and Assessment in Software Engineering Conference*, 2008.

[23] I. Hadar, I. Reinhartz-Berger, T. Kuflik, A. Perini, F. Ricca, and A. Susi, "Comparing the comprehensibility of requirements models expressed in Use Case and Tropos: Results from a family of experiments," *Information and Software Technology*, vol. 55, no. 10, pp. 1823–1843, 2013.

[24] V. A. Burégio, S. R. L. Meira, and E. S. Almeida, "Characterizing dynamic software product lines-a preliminary mapping study." in *14th International Software Product Lines Conference (SPLC)*, ser. SPLC '10 Workshops, 2010, pp. 53–60.

[25] N. Bencomo, J. Lee, and S. O. Hallsteinsen, "How dynamic is your dynamic software product line?" in *14th International Software Product Lines Conference (SPLC)*, ser. SPLC '10 Workshops, 2010, pp. 61–68.

[2] http://www.ines.org.br