A Preliminary Comparison of Using Variability Modeling Approaches to Represent Experiment Families

Amadeu Anderlin Neto[1,3], Marcos Kalinowski[1], Alessandro Garcia[1], Dietmar Winkler[2], Stefan Biffl[2]

[1] Dept. Informatics, SE Lab PUC-Rio
22451-900 Rio de Janeiro, Brazil
[aanderlin, kalinowski, afgarcia]@inf.puc-rio.br

[2] CDL-SQI, Information Systems Eng.
Vienna University of Technology
A-1040 Vienna, Austria
<firstname>.<lastname> @tuwien.ac.at

[3] Federal Institute of Education Science and Technology of Amazonas
69083-000 Manaus, Brazil
amadeu.neto@ifam.edu.br

ABSTRACT

Background: Replication is essential to build knowledge in empirical science. Experiment replications reported in the software engineering context present variabilities on their design elements, e.g., variables, materials. The understanding of these variabilities is required to plan experimental replications within a research program. However, the lack of an explicit representation of experiments' variabilities and commonalities is likely to hamper their understanding and replication planning. **Aims**: The goal of this paper is to explore the use of *Variability Modeling Approaches* (VMAs) to represent experiment families (i.e., an original study and its replications) and to investigate the feasibility of using VMAs to support experiment replication planning. **Method**: We selected two experiment families, analyzed their commonalities and variabilities, and represented them using a set of well-known VMAs: Feature Model, Decision Model, and Orthogonal Variability Model. Based on the resulting models, we conducted a preliminary comparison of using such alternative VMAs to support replication planning. **Results**: Subjects were able to plan consistent experiment replications with the VMAs as support. Additionally, through a qualitative analysis, we identified and discuss advantages and limitations of using the VMAs. **Conclusions**: It is feasible to represent experiment families and to plan replications using VMAs. Based on our emerging results, we conclude that the Feature Model VMA provides the most suitable representation. Furthermore, we identified benefits in a potential merge between the Feature Model and Decision Model VMAs to provide more details to support replication planning.

CCS CONCEPTS

• **Software and its engineering ~ Empirical software validation**

KEYWORDS

Experiment planning, experiment replication, experiment lines

ACM Reference format:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

EASE '19, April 15-17, 2019, Copenhagen, Denmark
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-7145-2/19/04…$15.00
https://doi.org/10.1145/3319008.3319356

1 INTRODUCTION

Experimental research is an important approach for advancing the maturation of the *Software Engineering* (SE) approaches [26]. Through experimentation, researchers can collect evidence related to the performance of a method or technology [2]. Furthermore, experiments need replications in different contexts, at different times, and under different conditions to gain generalizable knowledge [5][10]. According to Silva *et al.* [22], replication of empirical studies is an essential activity in the construction of knowledge. Moreover, replications increase the validity and reliability of original study results [10].

Through experimental studies we can verify how and when technologies work, understand their limitation and identify opportunities for improvement [2]. For replications, Shull *et al.* [21] present two implications: to gain confidence in results of previous studies (result reproduction) and to understand the sources of variability that influence a given result. There are guidelines (e.g., [6]) that can help the researchers to report experimental replications [13]. Nevertheless, these guidelines do not address the explicit representation of variabilities and do not support planning of experiment replications.

Considering the lack of variability modeling, we recently introduced the idea of using experiment lines to facilitate designing experiment replications in software engineering. In an experiment line designs are based on potential configurations of experiment elements [4]. In the area of *Software Product Line* (SPL), there are several *Variability Modeling Approaches* (VMAs) to support variability management [23] in different contexts. According to Czarnecki *et al.* [8] three existing VMAs have gained most importance: *Feature Model* (FM), *Decision Model* (DM), and *Orthogonal Variability Model* (OVM).

In this paper, we tackle the idea of using experiment lines to support planning of experiment replications. Therefore, we identified two representative experiment families, conducted by independent authors. We have selected these families as they concern different SE areas, and involve several replications each. We analyzed the variabilities among the replications and represented them using FM, DM, and OVM. We conducted a preliminary comparison in order to identify advantages and limitations of using each VMAs. Our goal is to understand to what extent they support experiment replication planning. Our emerging results indicate that VMA models may support the planning of experiment replications. Also, among different candidate VMAs, FM provides the most comprehensive overview. Nevertheless, we identified advantages and limitations for each of the models and, therefore, opportunities for future research.

The remainder of this paper is organized as follows. Section 2 provides the background. Section 3 introduces the research questions. Section 4 describes how the experiment families were represented using VMAs. Section 5 describes the preliminary evaluation and its results. Finally, Section 6 concludes and outlines future work.

2 BACKGROUND

In this section, we provide brief background on the problem space, replications in SE experiments, and on the solution space, VMAs.

2.1 Replications in SE Experiments

The term replication can be defined as an empirical study that addresses questions or hypotheses similar to those posed in previous studies [22]. Shull *et al.* [21] present two types of replications: (1) exact replications, following the procedures of an experiment as closely as possible; and (2) conceptual replications, evaluating the same research questions by using a different experimental procedure. For exact replications, there are two sub-categories: dependent replications, where all conditions of the experiment are the same or very similar; and independent replications, where researchers vary one or more experiment design elements.

Regarding the current state of replications in SE, Silva *et al.* [22] conducted a *Systematic Mapping Study* (SMS) and identified 133 replications of 72 original studies, from 1994 to 2010. Bezerra *et al.* [3] updated the earlier SMS. They found 51 replications of 35 original studies between 2011 and 2012. There are typically less than two replications per original study and this number seems to be increasing. Moreover, Magalhães *et al.* [13] analyzed investigations related to replications (e.g., recommendations, frameworks, guidelines). While understanding the sources of variability that influence a given result is one of the main implications of conducting replications [21], none of these investigations discusses variabilities in the context of experiment replication planning. Hence, there is an opportunity to investigate how to represent variabilities of experiment families. Furthermore, we believe VMAs can be used for variability modeling and to support planning experiment replications.

2.2 Software Product Lines and Variability Modeling Approaches

The main goal of the SPL engineering is to provide similar products with specific features, based on reusing a common architecture [15]. According to Altintas *et al.* [1], a SPL aims at establishing a product family through common and reusable components. SPLs provide means to efficiently design, produce, and maintain multiple similar software variants, exploiting their commonalities and managing their variabilities [15]. We identified similar challenges for experiment families in a research program, as their designs also involve commonalities and variabilities. Several VMAs have been proposed for managing both common and varying elements in SPL. In this paper, we selected three VMAs [8] with capabilities that may be relevant for representing experiment lines: FM, DM, and OVM. These models are described below.

Feature Model (FM) was proposed in 1990 as part of the *Feature-Oriented Domain Analysis* (FODA) method [11]. Nowadays, FM is widely used in the SPL context to represent static feature commonalities and variabilities, to represent dependencies between features, and to determine allowed or forbidden combinations of features. A number of extensions and variants of the original FODA notation have been proposed. In this work, we choose the *Cardinality-Based Feature Model* (CBFM), an extension of original FM that integrates several extensions of previous approaches [7]. The CBMF is more expressive to describe commonalities and variabilities by introducing the concept of cardinality [7].

Decision Model (DM). Most of existing DM approaches have been influenced by the Synthesis Method [24]. While the main purpose of FM is the domain, the DM emphasizes decisions in the process of product derivation, in order to distinguish among the members of a product family and to guide adaptation of work products. Furthermore, a decision model documents the decisions made to specify a member of a domain. This idea fit well with the tabular notation of DM presented in [8]. For the comparison presented in this paper, we selected the tabular notation to represent the DM.

Orthogonal Variability Model (OVM). OVM is a modeling approach proposed by Pohl *et al.* [15]. This approach is intended to support the variability management in SPLs by relating commonalities and variabilities to requirements, architecture, and other lifecycle artifacts. Only variabilities of the SPL are documented in this model. An OVM is composed of *Variation Points* (VP) and *Variants* (V). A VP is related to functionalities that can vary. A V, on the other hand, regards the possible instances of a VP. The notation used in this paper is based on [19].

As the three VMAs follow different concepts to represent variabilities that may be relevant for planning and analyzing an experiment line, we investigate the advantages and limitations of these VMAs to represent knowledge on experiment families.

3 RESEARCH QUESTIONS

The main goal of this paper is to investigate whether and how VMAs can be useful to represent experiment families. We also want to gather an initial understanding on if and how they can support the planning of experiment replications. Therefore, we discuss the following research questions. *RQ-1: How can software variability modeling approaches (VMAs) be used to represent experiment families?* and *RQ-2*: *How can VMA representations support planning experiment replications?* In order to answer these questions, we built models based on FM, DM and OVM notations and conducted a preliminary evaluation.

RQ-1. We believe that VMAs can help to keep the overview and consistency of the planned experiment family and add value to the design of experiment replications. VMAs typically describe rules for keeping consistency and can handle complex variability situations. Indeed, VMAs have been successfully applied in the SPL context and there are similar challenges concerning variabilities in SPLs and experiment families. Even small changes in a replication may lead to unexplained differences in the results. Thus, it is important to keep track of these changes. In the context of an experiment family, we relate features, decisions, or variation points to theory constructs used in the experiment design.

RQ-2. Regarding these theory constructs, Wohlin *et al.* [26] consider that replications may present variabilities of the following factors: (1) *Site* where the experiment is conducted; (2) *Experimenters* conducting the experiment; (3) *Design* chosen for the experiment; (4) *Instrumentation*, i.e., forms and other material; (5) *Variables* measured; and (6) *Subjects* conducting the experiment. The guidelines proposed by Carver [6] emphasize that changes related to the design, participants, artifacts, procedures, data collection, and/or analysis techniques should be clearly reported in replication papers. Based on these potential sources of variation, we built FMs, DMs, and OVMs for two representative experiment families and analyzed their advantages and limitations.

4 REPRESENTING EXPERIMENT FAMILIES USING VMAs

In order to address RQ-1 and investigate ways to represent experiment lines with FM, DM and OVM, we selected two experiment families based on a SMS on replications of empirical studies [3][22]. Our criterion was to select experiment families with solid experiment designs, addressing investigations in different SE areas, and with several replications. It is noteworthy that we did not analyze the quality and/or validity of the results of the experiment families. We only analyzed the structure of the experiment planning.

After identifying a set of candidates and internal discussion, we selected two experiment families, both with their original study published in a high-quality software engineering venue. The original study of an experiment family on *software requirements inspection* by Porter *et al.* [16] started 1995 and had five replications. The original study of the second experiment family on *code maintenance* by Prechelt *et al.* [17] was conducted in 1997 and had 6 replications. Due to space constraints, in this paper we will only present a fragment of the models for the second experiment family. The description of the first experiment family, the full models and the evaluation package are available online[1].

The original study from the family of experiments [17] (hereafter called *PatMain* experiment) aimed at comparing designs using patterns with alternative designs in the context of program maintenance. The *PatMain* experiment had six replications:

- Vokáč *et al.* [25] – one replication (FirstRep).
- Prechelt and Liesenberg [18] – one replication (SecondRep).
- Juristo and Vegas [9] – one replication (ThirdRep).
- Nanthaamornphong and Carver [14] – two replications (FourthRep and FifthRep).
- Krein *et al.* [12] – one replication (SixthRep).

The *PatMain* experiment was conducted with 29 unpaid professional software engineers from a single company. The subjects were divided into four groups (six to eight subjects in each group). Groups were balanced using random blocked assignment where professional experience, C++ experience, and the level of knowledge in relevant patterns were considered. The independent variables were: (1) the programs and change tasks; (2) the program version; and (3) the level of pattern knowledge. Time spent and correctness of solution were defined as dependent variables. Regarding the experiment object (*artifact*), there are four programs. Each program had two versions: with and without patterns. Subjects were asked to conduct maintenance tasks on the programs. One of the programs had three tasks, while the others had two.

The experiment was performed as follows. In the morning of day one, a pretest was conducted. The subjects had to carry out the maintenance tasks related to two programs. In the afternoon of day one and the morning of day two, a training course on design patterns was provided to participants. Finally, a post-test was conducted in which the professionals implemented solutions for the tasks of the other two programs. It is noteworthy that in each round the subjects used one program version with design patterns and another version without the design patterns.

Figure 1 shows the variabilities identified in the replications of *PatMain* experiment represented as a FM. The tabular DM notation is presented in Figure 2. Finally, the OVM can be seen in Figure 3. Each figure presents the same fragment of identified variabilities. Examples of identified variabilities to enable the understanding of each model are provided hereafter.

A source of variabilities was related to the independent and dependent variables. In ThirdRep the amount of pattern knowledge (independent variable) was not considered and task correctness (dependent variable) was not explored. Regarding subjects, researchers commonly apply questionnaires to characterize participants' background. They typically consider experience, knowledge and, in some cases, education. The *PatMain* experiment considered these three characteristics in order to classify subjects into different levels. Thus, the researchers were able to divide participants in balanced groups using random blocked assignment. However, FourthRep did not consider education to apply blocking of subjects. Furthermore, the groups were randomly assigned without blocking and balancing in the ThirdRep.

Another observed variability concerned training. A training phase was performed only in *PatMain* and FirstRep. It is noteworthy that SecondRep to SixthRep took part in a joint replication project. This project aimed at performing several replications of the *PatMain* experiment simultaneously. All replications belonging to the project conducted only the pretest and did not conduct training. However, we observed that training becomes mandatory when researchers consider the amount of pattern knowledge as an independent variable. Moreover, if there is training, the experiment should be conducted in two (not necessarily consecutive) days.

In summary, for a replication within the context of the code maintenance and using design patterns, the information provided by the VMAs could be interpreted as follows. In order to test the hypotheses, independent and dependent variables are defined. The experiment replication plan could reuse definitions of previous replications or introduce new variabilities. Regarding subjects, experience has been measured in the replications, while education has been optional. The training is not required, but if it is provided, the experiment should take two days.

In the following section, we present the results of a preliminary comparison of using FM, DM, or OVM to support planning an experiment replication. We also highlighted the advantages and disadvantages of using each of the models.

5 EVALUATION AND RESULTS

In order to answer the research questions RQ-1 and RQ-2, we conducted a preliminary evaluation with three independent subjects. A summary of their characterization follows. Subject 1 (S1) and Subject 2 (S2) have master degrees in software engineering and Subject 3 (S3) has a doctor degree in software engineering. All of them had previous experience with experiment planning. S1, S2, and S3 planned ten, four, and six experiments, respectively. Regarding replications, only S2 planned one. With respect to experience participating as subjects in experiments related to the inspection or maintenance domains, S1 participated in one experiment related to software inspection, while S2 and S3 participated in six each. Only S1 participated in an experiment concerning code maintenance. Finally, regarding experience with SPL, S2 has been working with SPLs as a team member in industry for four years, while the others had only studied it in a classroom context.

[1] www.inf.puc-rio.br/~kalinowski/ease2019

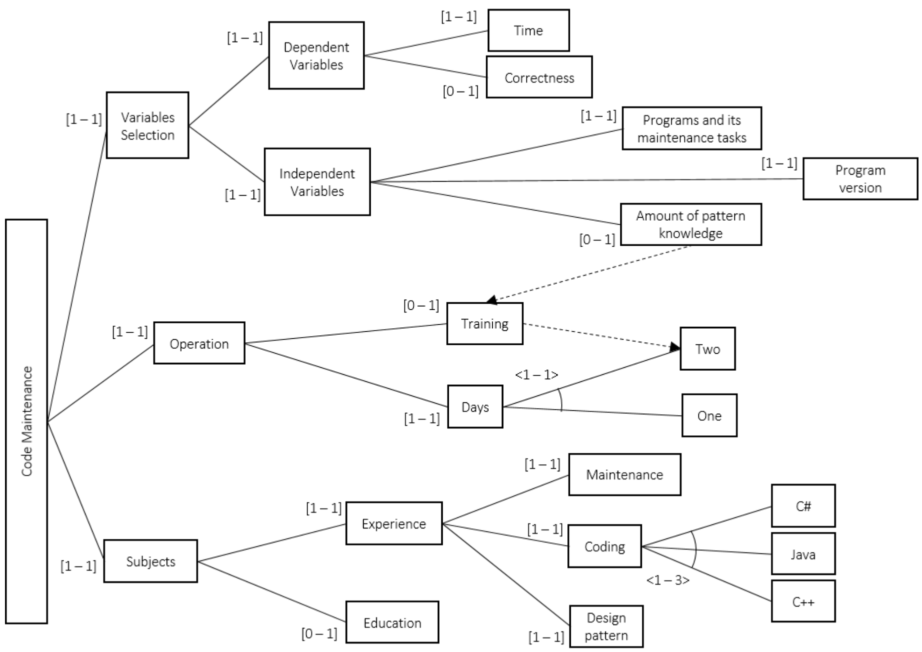

Figure 1: Fragment of the experiment family on Code Maintenance represented by a Feature Model.

Decision name	Description	Type	Range	Cardinality/constraint	Visible/relevant if
Correctness	Do you use the correctness as dependent variable?	Boolean	true | false		
Pattern_Knowledge	Do you use the amount of pattern knowledge as independent variable?	Boolean	true | false	If selected Training = true	
Training	Do you conduct training before experiment execution?	Boolean	true | false	If selected Days.Two = true	
Days	How many days to conduct the experiment?	Enum	One | Two	1:1	
Coding	Which programming language do the subjects have experience in coding?	Enum	C# | C++ | Java	1:3	
Education	Do you use subjects' education as metric to form groups?	Boolean	true | false		

Figure 2: Fragment of the experiment family on Code Maintenance represented by a Decision Model.

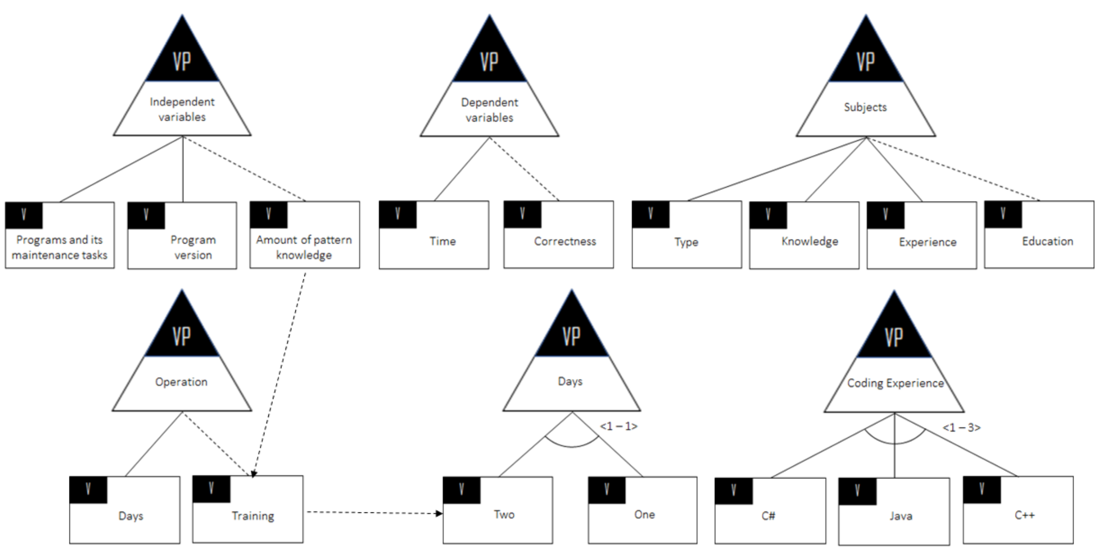

Figure 3: Fragment of the experiment family on Code Maintenance represented by an Orthogonal Variability Model.

Related to the evaluation design, each subject evaluated two models, one of each experiment family. We consider that FM presents more information than DM and OVM, because this model presents both commonalities and variabilities. Thus, the FM is only evaluated in Round 2. Because DM describes decisions, we consider that DM provides more information than OVM for S1. Therefore, the OVM was evaluated before DM when a subject had to evaluate these two models. The design of the evaluation is presented in Figure 4.

	Round 1	Round 2
Subject 1	*OVM (FamilyOne)*	*DM (FamilyTwo)*
Subject 2	*DM (FamilyOne)*	*FM (FamilyTwo)*
Subject 3	*OVM (FamilyTwo)*	*FM (FamilyOne)*

Figure 4: Evaluation Design.

In each round, we asked subjects to plan a new experiment replication for the experiment family, based on a specific VMA representation (hereafter referred to as model). Their plan should have at least the experiment replication components described in Section 3, presented by Wohlin *et al.* [26] and Carver [6]. In each round, we briefly explained the context of each experiment family. The subjects had to read and analyze the VMA model they received and to fill in an experiment planning form with their choices regarding the variabilities. After completing each round, the subjects answered a follow-up questionnaire with open questions (on their strategy, advantages, limitations, and improvement suggestions), allowing them to register their perceptions on the model used during the task.

Based on the resulting planning and on the questionnaire responses, we conducted a qualitative analysis. According to Seaman [20], qualitative methods facilitate achieving a comprehensive understanding of the phenomenon under investigation.

Experiment Replication Planning. First, we analyzed the variabilities that the subjects chose and wrote in the experiment planning form. The resulting planning of all subject was in accordance with the model's requirements, except for S1. We observed that S1 did not select a mandatory independent variable (*program version*) when using OVM. Despite of this, we consider the resulting planning to be acceptable with minor adjustments. An explanation could be that S1 had no previous experience working with SPLs, and OVM in particular. On the other hand, S3, who also did not have previous experience, was able to plan according to the represented requirements. It is noteworthy that when using DM and FM models, all subjects were able to plan their replications following all the model requirements.

Strategy to Use the Model. As far as the follow-up questionnaire is concerned, we were able to derive some interesting observations. First, concerning the strategy to use the model, S1 informed that he *"tried to map the variation points, in Round 1, and the 'decision name', in Round 2, to experiments components"*. On the other hand, S2 and S3 stated that they started from the overall experiment scenario depicted in the models and made choices of the variations according to their experience.

Advantages of VMAs. With respect to the perceived advantages. For S3, "*both models [OVM and FM] help to acquire an overview of the experiment family*". For S2, an important benefit related to use of both models (DM and FM) is that *"they allow reusing components, which could be beneficial mainly for novice researchers"*. Therewithal, S2 also stated, *"these models can represent best practices and generate new scenarios that we can use to expand an experiment family"*. Directly related to these advantages, S1 stated, *"the OVM model helps inserting the researcher in the experiment context"*. S1 also reported an advantage of using DM, *"The DM provided a better understanding of each variability, because of the decision description"*. Another advantage mentioned by S1 concerned effort reduction, "*both models reduce the time needed to plan a replication"*. Overall, based on the answers the main perceived benefits were related to providing a comprehensive overview on the experiment family and its variabilities and enabling reuse.

Limitations of VMAs. Regarding VMA limitations, S1 reported a lack of sequence when using OVM, and a lack of overview on elements of the replication plan when using OVM and DM. In fact, when using OVM the VPs are not necessarily connected to each other, the analysis of the VPs has to be done one by one. In the tabular notation, used to represent the DM, the model has only textual information, losing the graphical overview. Note that OVM and DM represent only variabilities, while FMs represent both variabilities and commonalities. This difference was highlighted by S3, who stated, *"the FM presents more details"*. Finally, S3 stated as a limitation, *"it would have been important to know more details about the experiment family and, mainly, the artifacts that were used"* and that *"both OVM and FM can cause novice researcher and/or graduate students to not think outside the box"*. We agree that researchers should have a detailed look at least at the original experiment plan and instruments before planning replications. Thereafter, it would be easier to understand the variabilities represented in the model. Regarding not thinking out of the box, from our point of view this depends on how people are instructed to use the models, as they should keep in mind that they could add new variations in their specific replication to fit their purpose. Moreover, we emphasize that researchers will still need to analyze the complete resulting experiment replication plan to ensure the choices make sense. Indeed, as mentioned before, even though we have based our example on Prechelt *et al*. [17][18], we did not analyze the quality of this experiment family nor the validity of its results. Thus, readers considering pattern-based experiments should not copy elements of the experimental design without proper scientific reasoning about the decisions taken.

Suggestions for Improvement. Subjects also provided some suggestions for improvement. In order to improve the model represented by OVM, S1 suggested including goals and threats to validity. However, we disagree, mainly because the experiment goals are not very different, containing only minor changes. Regarding threats to validity, each threat depends on the whole experiment planning. Thus, it is not feasible to consider threats as part of a VMA model in this stage. A suggestion of S2 was to merge the DM with FM, like a mental map, and to develop a tool that incorporates the models. Finally, S3 suggested a collaborative tool with links to the artifacts. Many of these suggestions are in line with our future work, which is briefly discussed in the next section.

The main limitations of our preliminary comparison are as follows. The first one is related to the number of subjects. Only three subjects took part in the evaluation. However, we focused on qualitative analyses. No quantitative analysis was performed. The other limitation regards the order of use the VMAs. We consider using FM only in Round 2 and, in case of S1, DM is used after OVM. We believe that there were limited learning effects, since the VMAs used in each round represented different families and the models have significant differences between them. However, future evaluations will involve different designs and involve more subjects.

6 CONCLUSION AND FUTURE WORK

We were able to represent the experiment replication variabilities using the VMAs for both selected experiment families (RQ-1). Our preliminary comparison allowed observing that all three investigated VMAs (FM, DM and OVM) are useful for easily identifying variabilities and elements to reuse when planning experiment replications (RQ-2). However, there was a consensus among the study participants that the FM provides a more comprehensive overview, mainly because its representation is more complete by also providing information about commonalities.

Based on these emerging results, we believe that FM could be used as a starting point to represent experiment families and that using experiment lines to support planning experiment replications might be a promising direction of research. There are solutions applied in the SPL area (e.g., related to managing variability, maximizing reuse, supporting product derivation) that could potentially bring benefits when applied to experiment replications.

Our future work involves further investigating the use of VMAs to represent experiment families. In addition, we want to conduct further investigations towards conceiving a (possibly tool supported) solution that incorporates a VMA model and the experimental artifacts. We believe that such solution could enable employing systematic reuse to leverage evolving software engineering knowledge through experiment replications. In addition, we believe the tool can support not only new replications, but also opportunize new experiments, since the gaps to be researched will be easier to see.

Acknowledgment

The financial support by the Christian Doppler Research Association, the Austrian Federal Ministry for Digital and Economic Affairs and the National Foundation for Research, Technology and Development is gratefully acknowledged.

REFERENCES

[1] N. I. Altintas, S. Cetin, A. H. Dogru, and H. Oguztuzun. 2012. Modeling product line software assets using domain-specific kits. *IEEE Transactions on Software Engineering (TSE)*, 38(6), 1376-1402. doi: 10.1109/TSE.2011.109.

[2] V. R. Basili. 1996. The role of experimentation in software engineering: past, current, and future. *Proceedings of IEEE 18th International Conference on Software Engineering*, Berlin, 442-449. doi: 10.1109/ICSE.1996.493439.

[3] R. M. Bezerra, F. Q. da Silva, A. M. Santana, C. V. de Magalhães, and R. E. Santos. 2015. Replication of empirical studies in software engineering: an update of a systematic mapping study. *ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, Beijing, 1-4. doi: 10.1109/ESEM.2015.7321213.

[4] S. Biffl, M. Kalinowski, and D. Winkler. 2018. Towards an experiment line on software inspection with human computation. In *6th IEEE/ACM International Workshop on Conducting Empirical Studies in Industry, CESI@ICSE*.

[5] D. T. Campbell, and J. C. Stanley. 1963. Experimental and quasi-experimental designs for research.

[6] J. C. Carver. 2010. Towards reporting guidelines for experimental replications: a proposal. In: *RESER'2010: Proceedings of the 1st International Workshop on Replication in Empirical Software Engineering Research*.

[7] K. Czarnecki, S. Helsen, and U. Eisenecker. 2004. Staged condiguration using feature models. In R. L. Nord, editor, Software Product Lines: Third International Conference, Boston, MA, USA. Heidelber, Germany. Springer-Verlag Proceedings, volume 3154 of Lecture Notes in Computer Science, 266-283.

[8] K. Czarnecki, P. Grünbacher, R. Rabiser, K. Schmid, and A. Wasowski. 2012. Cool features and tough decisions: a comparison of variability modeling approaches. Proceedings of the Sixth International Workshop on Variability Modeling of Software-Intensive Systems, 173-182. doi: 10.1145/2110147.2110167.

[9] N. Juristo, and S. Vegas. 2011. Design patterns in software maintenance: an experiment replication at UPM – experiences with the RESER'11 Joint Replication Project. *2011 Second International Workshop on Replication in Empirical Software Engineering Research*, Banff, AB, 7-14. doi: 10.1109/RESER.2011.8.

[10] N. Juristo, and O. S. Gómez. 2012. Replication of software engineering experiments. In *Empirical Software Engineering and Verification*, Bertrand Meyer and Martin Nordio (Eds.). Springer-Verlag, Berlin, Heidelberg, 60-88.

[11] K. Kang, S. Cohen, J. Hess, W. Nowak, and S. Peterson. 1990. Feature-oriented domain analysis (FODA) feasibility study. Technical report, CMU/SEI-90TR-21.

[12] J. L. Krein, L. J. Pratt, A. B. Swenson, A. C. MacLean, C. D. Knutson, and D. L. Eggett. 2011. Design patterns in software maintenance: an experiment replication at Brigham Young University. *2011 Second International Workshop on Replication in Empirical Software Engineering Research*, Banff, AB, 25-34. doi: 10.1109/RESER.2011.10.

[13] C. V. de Magalhães, F. Q. da Silva, R. E. Santos, and M. Suassuna. 2015. Investigations about replication of empirical studies in software engineering: a systematic mapping study. *Information and Software Engineering*, 64, 76-101. doi: 10.1016/j.infsof.2015.02.001.

[14] A. Nanthaamornphong, and J. C. Carver. Design patterns in software maintenance: an experiment replication at University of Alabama. *2011 Second International Workshop on Replication in Empirical Software Engineering Research*, Banff, AB, 15-24. doi: 10.1109/RESER.2011.11.

[15] K. Pohl, G. Böckle, and F. J. van der Linden. 2005. Software product line engineering: Foundations, principles and techniques. Berlin, Heidelberg, New York: Springer.

[16] A. A. Porter, L. G. Votta, and V. R. Basili. 1995. Comparing detection methods for software requirements inspections: a replicated experiment. In *IEEE Transactions on Software Engineering*, 21(6), 563-575. doi: 10.1109/32.391380.

[17] L. Prechelt, B. Unger, W. F. Tichy, P. Brossler, and L. G. Votta. 2001. A controlled experiment in maintenance: comparing design patterns to simpler solutions. In *IEEE Transactions on Software Engineering*, 27(12), 1134-1144. doi: 10.1109/32.988711.

[18] L. Prechelt, and M. Liesenberg. 2011. Design patterns in software maintenance: an experiment replication at Freie Universität Berlin. *2011 Second International Workshop on Replication in Empirical Software Engineering Research*, Banff, AB, 1-6. doi: 10.1109/RESER.2011.12.

[19] F. Roos-Frantz, D. Benavides, A. Ruiz-Cortés, A. Heuer, and K. Lauenroth. 2012. Quality-aware analysis in product line engineering with the orthogonal variability model. Software Quality Journal, 20(3-4), 519-565.

[20] C. B. Seaman. 1999. Qualitative methods in empirical studies of software engineering. IEEE Transactions on Software Engineering, 25(4), 557-572.

[21] F. J. Shull, J. C. Carver, S. Vegas, and N. Juristo. 2008. The role of replications in empirical software engineering. *Empirical Software Engineeringn*, 13(2), 211-218. doi: 10.1007/s10664-008-9060-1.

[22] F. Q. da Silva, M. Suassuna, A. C. França, A. M. Grubb, T. B. Gouveia, C. V. Monteiro, and I E dos Santos. 2014. Replication of empirical studies in software engineering research: a systematic mapping study. *Empirical Software Engineering*, 19(3), 501-557. doi: 10.1007/s10664-012-9227-7.

[23] M. Sinnema and S. Deelstra. 2007. Classifying variability modeling techniques. Information & Software Technology, 49(7), 717–739.

[24] Software Productivity Consortium Services Corporation. 1993, Technical Report SPC-92019-CMC. *Reuse-Driven Software Processes Guidebook*, Version 02.00.03.

[25] M. Vokáč, W. Tichy, D. I. K. Sj□berg, E. Arisholm, M. Aldrin. 2004. A controlled experiment comparing the maintainability of programs designed with and without design patterns – a replication in a real programming environment. *Empirical Software Engineering*, 9(3), 149-195. doi: 10.1023/B:EMSE.0000027778.69251.1f.

[26] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslen. 2012. Experimentation in software engineering. Springer-Verlag Berlin Heidelberg.