

Evaluating the Representation of User Interface Elements in Feature Models: an Empirical Study

Ildevana Poltronieri Rodrigues*, Ana Paula Terra Bacelo*, Milene Selbach Silveira*,
Márcia de Borba Campos*, Elder de Macedo Rodrigues†

*School of Computer Science - Pontifical Catholic University of Rio Grande do Sul (PUCRS) – Porto Alegre, Brazil

†Federal University of Pampa – Alegrete, Brazil

Email: ildevana.rodrigues@acad.pucrs.br {ana.bacelo, milene.silveira, marcia.campos}@pucrs.br
eldermr@gmail.com

Abstract—Feature models are considered a reference in the majority of Software Product Line (SPL) methods. There are several different feature model notations to represent requirements of an SPL, at a higher or lower level of abstraction. Some notations present properties to distinguish functional, conceptual and non-functional features of an SPL. Non-functional requirements, especially those that involve the construction of user interfaces (UI), are usually not represented in the feature models, since the user interfaces are often created manually for each product. In this paper we present an experimental study performed in order to evaluate the effort required to, as well as the benefits and drawbacks of representing UI elements during the feature modeling of a Financial SPL. To this end, we applied the Odyssey-Fex and UI-Odyssey-Fex notations to design feature models from the perspective of Domain Engineers in the context of undergraduates, M.Sc. and Ph.D. students and software engineers with some expertise in SPL. Our results indicate that the effort to use the notations are similar, but the use of the UI-Odyssey-Fex notation provides a better representativeness of UI elements.

I. INTRODUCTION

A feature model (FM) represents all of the features, and the relationships between them, of a software product line (SPL) domain [1] [2] [5] [6]. A feature can be mandatory (must be present in all of the derived products) or optional (one or more domain products have this feature). A derived product is composed by the mandatory features and a selection of optional features.

There are different feature model notations, with an assorted number of elements, to represent the features of the SPL domain at different levels of abstraction [7]. Some notations present properties to distinguish functional, conceptual and non-functional features of an SPL. However, beyond these, in the construction process of a feature model for an SPL, user interface elements can be modeled in order to identify mandatory, optional and configurable interfaces among the different products of an SPL. Although there are several researches dealing with non-functional properties [13], those involved in the construction of user interfaces are normally not represented in the feature models or are not addressed in the context of the construction of an SPL. Pleuss et al. [11] point out that, in practice, the user interfaces are more often than not

created manually for each product, which impacts the usability, efficiency and maintenance of the product, without considering the aspects of cost involved in their construction.

There are different approaches for including usability aspects in software product lines, from their manual construction, as previously mentioned, to extending existing methods for software product lines, to include non-functional requisites [9] in possibilities of customization based on abstract models [4]. Using abstract models and even extending already existing models is very costly, with the team having to develop (and/or extend) one more of many models, which ends up being unfeasible considering issues regarding the necessary cost, time and human resources.

Although there are many FM modeling notations available, and a few of which supporting UI elements, SPL engineers face some doubts when considering to model UI elements into a FM, such as: Does an UI-supporting notation present some advantage regarding ease of use when compared with a standard FM notation?; What is the effort required to create a FM with UI elements? Does the use of an UI-supporting notation provide a better representativeness of UI elements?. Despite these doubts, to the best of our knowledge there is no work discussing the required modeling effort, benefits and issues on the use of notations that support the representation of UI elements. Motivated by this lack of knowledge, in this paper we report an experimental study to provide evidence about the advantage, the required effort to design the models and representativeness of UI elements.

This paper is organized as follows. Section 2 presents the experiment instruments, the FM notations, the SPL requirements documents and our design guidelines to model the feature diagrams. Section 3 provides details about the experimental design and Section 4 describes the execution of the experiment. Section 5 presents our analysis on achieved results. We conclude the paper in Section 6 with final considerations and some future research directions.

II. EXPERIMENT INSTRUMENTS

In this section we briefly introduce the Odyssey-Fex and UI-Odyssey-Fex feature model notations.

A. Odyssey-Fex

The Odyssey-Fex notation [3] is used in the Odyssey reuse environment [10]. Different from the notations proposed by

Favaro and Mazzini [4], Gurp et al. [15] and Czarnecki et al. [2], this notation also indicates that the features should be classified into different types: Domain, Entity, Operational, Domain Technology and Implementation Techniques. The Odyssey-Fex notation also proposes a different way of representing the relationship between features. As such, it uses the representation of UML relationships and the specific notation relationships, as proposed by Miller [8]. In addition to the different types of features and their relationships, the variability aspects of an SPL in the OdysseyFex notation are also considered. As such, features can be specified as mandatory (they exist in all of the products of an SPL), optional (they may or may not exist in one or more of the products of a line) or variation points (they are configurable considering the product of which it is part). Figure 1 shows an example of feature model created in accordance with the Odyssey-Fex notation, for the domain of Financial Institutions. The Financial domain is represented by the conceptual feature *Financial*, with which the actors *Customer* and *Employee* interact. This feature is associated with the functional feature *Authenticate*. The *Authenticate* feature is implemented by several other functional features, such as *Credit Card Balance* and *Transfer*. The mandatory *Transfer* feature must be implemented by the *Checking to Saving*, *Saving to Checking*, *Accounts from Others Banks* and *Accounts from Same Bank*. As can be seen in the figure, there is no element of this notation to explicitly represent UI components in the FM.

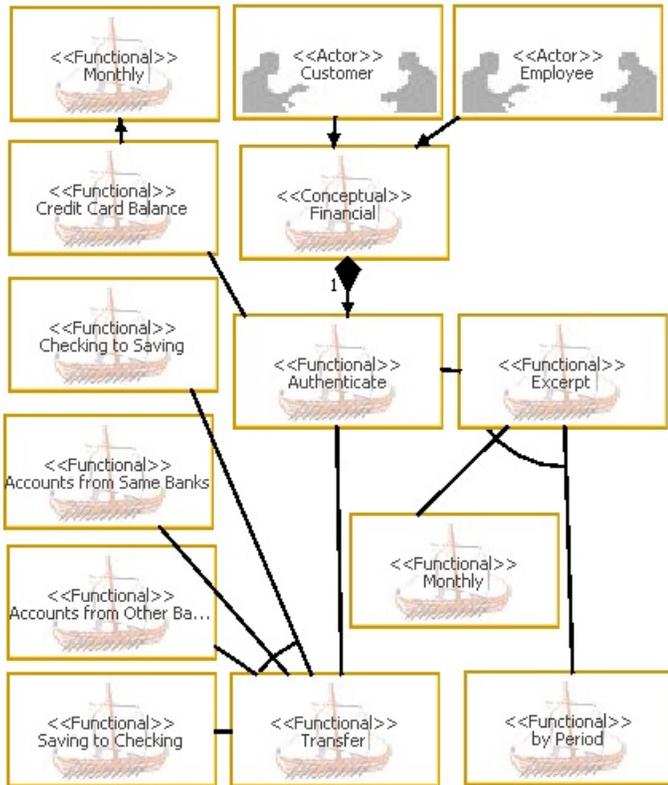


Fig. 1. Feature Model – Odyssey-Fex notation

B. UI-Odyssey-Fex

To mitigate the Odyssey-Fex limitation in representing UI elements an extension, called UI-Odyssey-Fex, was developed.

This extension provide a feature element to model features that represent aspects of user interface [11]. In UI-Odyssey-Fex, all elements need to represent features in the original notation were implemented and, in addition to this, a new feature category was included, called User Interface. An example of the FM designed in accordance with the extension is presented in Figure 2. The figure shows the same SPL represented in Section II-A, but including the User Interface features category. In this case, the SPL engineer could represent the UI elements that are shared among the Financial products derived from the SPL. For instance, the *Transfer* functional feature has a communication link with the User Interface *Transfer System* feature.

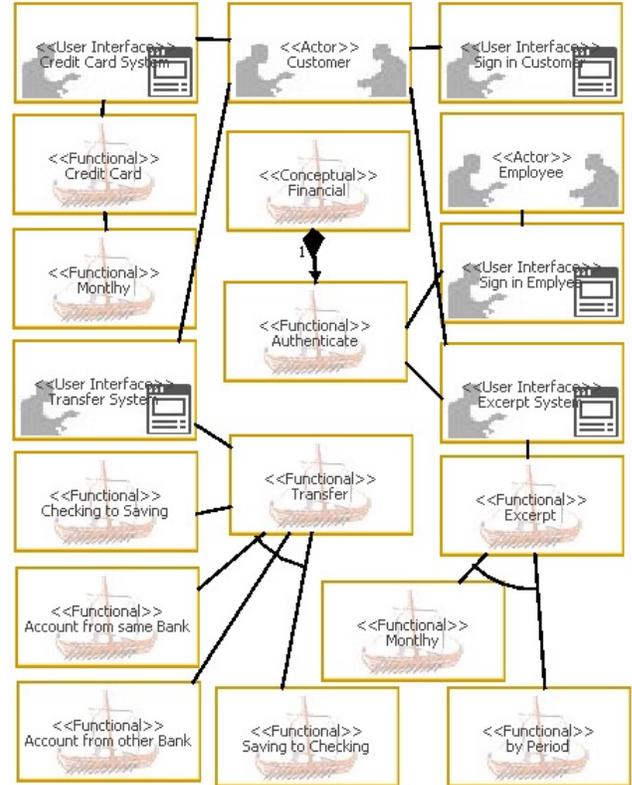


Fig. 2. Feature Model – UI-Odyssey-FEX notation

III. EXPERIMENT DESIGN

In this section we introduce the Research Questions (RQ), present our experimental design, describe the experiment instruments and discuss the main threats to the study validity of the study:

RQ1: *Does the UI-Odyssey-Fex notation present advantages regarding ease of use, when compared with the Odyssey-Fex notation?*

Null Hypothesis H_0 : There is no advantage, regarding ease of use, when using Odyssey-Fex or UI-Odyssey-Fex notations. **Alternative Hypothesis H_1 :** The use of UI-Odyssey-Fex notation presents some advantage regarding ease of use when compared with the Odyssey-Fex notation. **Alternative Hypothesis H_2 :** The use of Odyssey-Fex notation presents some advantage regarding ease of use when compared with the UI-Odyssey-Fex notation.

RQ2: Does the UI-Odyssey-Fex notation require less effort (time spent) to design a feature model, when compared with the Odyssey-Fex notation?

Null Hypothesis H_0 : The effort is the same when using Odyssey-Fex or UI-Odyssey-Fex notations to design a feature model. **Alternative Hypothesis H_1 :** The use of the UI-Odyssey-Fex notation requires less effort to design a feature model, when compared with the Odyssey-Fex notation. **Alternative Hypothesis H_2 :** The use of the Odyssey-Fex notation requires less effort to design a feature model, when compared with the UI-Odyssey-Fex notation.

RQ3: Does the UI-Odyssey-Fex provide a better representativeness of UI elements, when compared with the Odyssey-Fex notation?

Null Hypothesis H_0 : The representativeness is the same, when using the Odyssey-Fex or UI-Odyssey-Fex notations, to represent UI elements. **Alternative Hypothesis H_1 :** The use of the UI-Odyssey-Fex notation better represents UI elements when compared with the Odyssey-Fex notation. **Alternative Hypothesis H_2 :** The use of the Odyssey-Fex notation better represents UI elements when compared with the UI-Odyssey-Fex notation.

A. Design

In our experiment we used an in-vitro approach, since all the experiment sessions were executed in a laboratory, under controlled conditions and using documentation from a toy example SPL. As subjects, we invited undergraduate, master and Ph.D. students from a university; from a company we invited software engineers with some expertise in SPL. The students were from Computer Science and Information Systems courses. It is important to highlight that each student had a different level of knowledge on SPL concepts, FM design and UI element representation. For instance, some students had some experience as a software analyst or as a developer, or experience designing and developing software during an IT course. The professional subjects were from an IT company, with an advanced level of knowledge about SPL modeling, but with an introductory knowledge on UI concepts. It is important to note that, initially, all the subjects answered a survey, thus enabling us to classify them into three blocks (beginner, intermediate and advanced), according to their background in software engineering - such as software design with UML and software product line concepts (see Figure 3). After that, the subjects were randomly (randomized block design [16]) allocated in two groups: Group 1 and Group 2. Subjects in Group 1 started modeling a SPL using the UI-Odyssey-Fex notation and then using the Odyssey-Fex notation to create an equivalent model. Conversely, Group 2 started modeling a SPL using the Odyssey-Fex notation and then applied the UI-Odyssey-Fex notation to create an equivalent model. Hence, all the subjects executed both treatments (Odyssey-Fex and UI-Odyssey-Fex notations). Moreover, we also *balance* the groups, so a similar number of subjects started modeling a feature model with a different notation (Group 1 with 16 subjects and Group 2 with 15 subjects). Figure 3 summarizes the experiment phases.

B. Instrumentation

During the experiment design and execution phases several *objects* were generated/used. The main experiment objects are the FM diagrams designed by the subjects to model a financial SPL - one diagram for each notation. Moreover, other objects were prepared and provided to support the subjects during the execution of the experiments tasks, such as training material, applications requirements, tools manual and examples of feature model diagrams. To create the features models we have provided a modeling supporting tool, called Odyssey [10]. This tool and the notations were presented to the experiment subjects during the training sessions, where we provided a manual with detailed instructions on how to apply the modelling tools to design the feature models in accordance with each notation. These sessions were conducted in a controlled laboratory, and all the experiment's subjects attended simultaneously. During the training sessions, the subjects modeled an SPL for the Retail domain, and they could ask open questions about the application documentation, notations and tools. These questions and their answers were shared among all the subjects. In the experiment execution phase the subjects modeled, using both modeling notations, an SPL for the Financial domain, based on its documentation. The modeling tool used were the same and all the subjects used the same computational resources. Moreover, the laboratory was isolated to avoid interruptions and any communications among the subjects or with the exterior world (e-mail, instant message, cellphones, etc) were not allowed. During the execution we collected effort metrics for each subject, as well as collected qualitative data after every session through a questionnaire.

C. Threats to Validity

Throughout the experimental process we identified some threats to the validity of our experimental study. In this section we describe these threats, as well as how we worked to mitigate them. To classify the threats we adopted the classification scheme proposed by [16]:

- **Conclusion validity:** we had identified some threats that could affect the ability to draw conclusions about relations between the treatment (notations) and the outcome of our experiment. One of the major threats is the small sample of subject (31 subjects), and the heterogeneity of the subjects profiles. They had a different level of expertise on software engineering, software product lines and modeling, and their different academic degrees. To mitigate these threats we took into consideration their academic degree and professional expertise to categorize them into blocking variables (beginner, intermediate and advanced).
- **Internal validity:** to mitigate the threats to the internal validity of the experiment we took the following decisions during the experiment design: the experiment session (training and execution) dates were defined to avoid periods in which subjects may be exposed to external influences, e.g., avoid running the experiment during the exam period (student subjects) and close to the start/end of a subject's project (industry subjects).

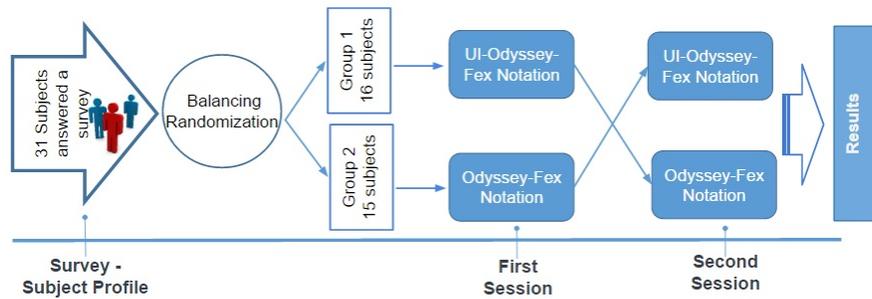


Fig. 3. Experiment Phases

- External validity:** we were aware that the selected sample of subjects may be a threat to the external validity of our study, since those subjects may not represent the SPL modeling professionals (population). To mitigate this threat, we only selected students and professionals with skills/expertise in this domain, such as SPL concepts and software modeling. Another threat to our study was that since the subjects learn the Odyssey-Fex notation, it could be easier to model with UI-Odyssey-Fex (it is an extension of Odyssey-Fex notation). This learning curve may influence the results. To mitigate this threat, the subjects of each block were randomly divided into two groups, where each group started modeling with a different notation. Another threat is that the application documentation may not be clear and unambiguous, leading the subject to misunderstand the SPL domain, its features and their relations, as well as its UI elements. To mitigate this threat we invited an expert in software product lines, software reuse and modeling, and another expert in Human Computer Interaction to review the experiment artifacts.
- Construct validity:** it may occur that a subject can erroneously conclude that their personal performance is measured to rank the experiment subjects or decide to respond favorably to the researcher. For instance, a student may be inclined to answer positively a survey of his professor. To mitigate these threats, before each session a researcher explained that the focus of the study was to evaluate modeling notations, not the subjects. Moreover, we clearly stated that we were not involved in the development of any of the notations, so we do not have any preference. The correct analysis and statistical interpretation of the results may be another threat to experiment construct validity. To mitigate this threat we invited an expert in statistical analysis from the Faculty of Mathematics of PUCRS.

IV. OPERATION OF THE EXPERIMENTAL STUDY

In this section we present in detail the preparation and execution activities performed during the experiment operation.

A. Preparation

In the preparation, we present how the experiment was conducted, how the documentation was prepared and how the experiment environment was configured. We also present how the experiment subjects were involved and motivated.

Our first personal contact with the subjects was made through an introductory presentation session, where an initial explanation

about the general idea of the experiment was presented. At this first moment we also provided a time slot, where the subjects could ask open questions about the experiment setup as well as about SPL and the FM notations and tools. At this moment we also discussed the research goal and how the subjects' data (e.g. experiment effort data) would be published. During this first session we presented the basics of software product lines, FODA [5] and FeatuRSEB [15] methods, as well as the Odyssey-Fex and UI-Odyssey-Fex notations.

TABLE I. EXPERIMENT SUBJECT DISTRIBUTION

| Groups | Blocking | | Number of subjects |
|-------------------|----------|--------------|--------------------|
| Group 1 | Students | Beginner | 6 |
| | | Intermediate | 2 |
| | | Advanced | 1 |
| | Industry | Beginner | 5 |
| | | Intermediate | 2 |
| | | Advanced | 0 |
| Group 2 | Students | Beginner | 3 |
| | | Intermediate | 4 |
| | | Advanced | 1 |
| | Industry | Beginner | 6 |
| | | Intermediate | 1 |
| | | Advanced | 0 |
| Total of Subjects | | | 31 |

During the second session, performed in a different date, manuals were presented to guide the use of the Odyssey-Fex and UI-Odyssey-Fex notations and tools. At this time, all the participants received instructions on the semantics of models, as well as a detailed demonstration of how to use the tools. At the final part of the session it was presented an example of a requirements document of a retail SPL (similar to a MD-50 functional requirement document) and its correspondent feature model. Based on this, it was possible to discuss and answer questions from participants on the notations' syntax, such as on the use of tools. At the end of the session a subject profile form was provided to all the subjects. The information extracted from the forms was used to classify and distribute the subjects through the blocks before running the experiment. To classify a subject into the blocks we used an ordinal scale for each response of a multiple choice questionnaire [16]. For instance, the subjects' experience with SPL concepts and feature modeling design was measured on a four level ordinal scale: 1 = none, 2 = beginning level (studied in a class), 3 = intermediate level (practised in a class project or in a small industry project), 4 = advanced level (used in more than one industrial project). Based on this, we evaluated the average experience of each subject and used it to distribute them into the blocks. The distribution of the industrial and student subjects into the blocks is depicted in Table I.

B. Execution

The experiment execution phase was comprised of two sessions (blue boxes in Figure 3): Session one - was used by the subjects from the Group 1 to design the FM using the UI-Odyssey-Fex notation, while subjects from Group 2 designed the FM using the Odyssey-Fex notation. It is important to note that to avoid misunderstanding, we provided the requirement documents with only those requirements, seven requirements in total, that the subject must design. Both groups used the same application documents: requirements and models. Moreover, they also designed the models at the same time, but they used different and isolated lab rooms. Session two - was used by the Group 1 subjects to design the FM using the Odyssey-Fex notation, while subjects from Group 2 designed the FM using the UI-Odyssey-Fex notation. As in the first session, both groups, separately, applied the same application documents that had been provided previously in session one. It is important to highlight the sessions one and two were performed in different dates. To monitor the time spent by each subject, the researcher who ran the experiment used a software to control the starting and ending time of each subject.

At the end of each session, all the subjects, from both groups, answered the same survey, to evaluate the ease of use and representativeness of each notation. For each survey question we provided three close-ended options (Linkert scale): *Disagree*, *Neither agree nor disagree* and *Agree*.

V. RESULTS

In this section, we present the data collected from our experiment, as well as summarizing our general findings. We performed hypothesis testing for all data sets using the R Studio statistical environment for R [12]. First, we evaluated the influence of the order in which the subjects applied the treatments (group one or group two), with the Chi-Square (χ^2) normality test [14], considering a significance level $\alpha = 0.05$. The results for the data collected from both notations shows that the difference between the observed and expected frequencies in each category are very small (see Table II), allowing us to reject the assumption that the order in which the subjects apply the notation did not affect the results.

TABLE II. CHI-SQUARE NORMALITY TEST

| Treatments | χ^2 | α |
|----------------|----------|----------|
| UI-Odyssey-Fex | 1.9273 | 0.4658 |
| Odyssey-Fex | 2.6467 | 0.3233 |

A. Hypothesis Testing

Since the order in which the subjects applied the treatments does not influence the results, we applied parametric Student's t-test for paired samples, considering a significance level $\alpha = 0.05$ and 30 degrees of freedom. In the remainder, we present and discuss the results per survey question, which in turn are related to the RQs stated in Section III. To evaluate the subjects' answers, each survey question has a ordinal scale for each response [16]. Therefore, the subjects must choose an answer that is measured on a three level ordinal scale: (1) agree, (0) partially agree and (-1) disagree. We also performed hypothesis testing applying t-Test (for paired samples), considering a significance level $\alpha = 0.05$ and 30 degree of freedom.

TABLE III. SUBJECTS ANSWERS - RQ 1

| Subjects Answers | | | | | | | | |
|------------------|-------------|---------|-------|-------|----------------|---------|-------|-------|
| Scale | Odyssey-Fex | | | | UI-Odyssey-Fex | | | |
| | Group 1 | Group 2 | Total | % | Group 1 | Group 2 | Total | % |
| Agree | 8 | 4 | 12 | 38.71 | 7 | 8 | 15 | 48.39 |
| Partially Agree | 5 | 9 | 14 | 45.16 | 5 | 6 | 11 | 35.48 |
| Disagree | 3 | 2 | 5 | 16.13 | 4 | 1 | 5 | 16.13 |

RQ1: *Does the UI-Odyssey-Fex notation present advantages regarding ease of use, when compared with the Odyssey-Fex notation?*

This RQ is represented by a closed multiple choice question in both subjects surveys: *Does the (UI-Odyssey-Fex/UI-Odyssey-Fex) notation present advantages regarding ease of use?*. As shown in Table III, 38.71% of the subjects agreed that the Odyssey-Fex is easier to use, while 45.16% partially agree and 16.13% disagree with the affirmation. In turn, when design the models with the UI-Odyssey-Fex, 48.39% of the subjects agreed that the notation is easier to use and 16.13% disagree. This basic analysis indicate that UI-Odyssey-Fex is easier to use than Odyssey-Fex. However, we performed hypothesis testing in order to accept or reject the H_0 hypothesis. For the data set from the question we applied the t-Test to the paired sample, in this way comparing the answers for Odyssey-Fex and UI-Odyssey-Fex. The results are shown in Table IV. The results of the test was **t-Test = 0,8278**, which is smaller than the tabulated value (from the t-table [16]) which is **t= 2,0423**. Therefore, we **accept** the H_0 hypothesis and conclude that there are no differences, regarding ease of use, when using Odyssey-Fex or UI-Odyssey-Fex notations.

TABLE IV. PAIRED T-TEST QUESTION 1 - EASINESS OF USE

| Treatments | Paired t-Test | Degrees of Freedom | p-value |
|---------------------------------|-------------------|--------------------|---------|
| Odyssey-Fex e UI-Odyssey-Fex | 0.8278 | 30 | 0.4143 |
| Confidence Intervals 95% | - 0.1419 - 0.3355 | | |
| Mean of difference | 0.0967 | | |

RQ2: *Does the UI-Odyssey-Fex notation require less effort (time spent) to design a feature model, when compared with the Odyssey-Fex notation?*

To answer this RQ we collected the subjects' effort data, in minutes, per notation. Based on this data, we applied the t-Test for paired samples. As Table V shows, the result of the test was t-Test = -1.1803, which is smaller than the tabulated value of t= 2.0423 and a p-value of 0.2471 (which is greater than 0.05). Based on that we can **accept** the H_0 hypothesis and conclude that the effort is similar when using the Odyssey-Fex or UI-Odyssey-Fex notations to design a feature model.

TABLE V. PAIRED T-TEST - EFFORT DATA

| Treatments | Paired t-Test | Degrees of Freedom | p-value |
|---------------------------------|------------------|--------------------|---------|
| Odyssey-Fex e UI-Odyssey-Fex | -1.1803 | 30 | 0.2471 |
| Confidence Intervals 95% | -9.8641 - 2.6383 | | |
| Mean of difference | -3.6129 | | |

RQ3: *Does the UI-Odyssey-Fex provide a better representativeness of UI elements, when compared with the Odyssey-Fex notation?*

To answer this RQ we collected the data from two of the evaluation survey questions that refer to the syntax and semantics of UI elements represented in the notations. Based on this data, we applied the t-Test for paired samples. As Table VI shows, regarding to the syntax question, the result of the test was $t\text{-Test} = 3.8676$, which is higher than the tabulated value of $t=2.0423$ and $p\text{-value}$ of 0.0005, that is, significantly lower than the expected 0.05. Based on that we can **refute** the H_0 hypothesis and **accept** the H_1 hypothesis, concluding that the UI-Odyssey-Fex notation has adequate syntax to represent UI elements when compared with the Odyssey-Fex notation.

TABLE VI. PAIRED T-TEST QUESTION 3 - SINTAXE

| Treatments | Paired t-Test | Degrees of Freedom | p-value |
|---------------------------------|-----------------|--------------------|---------|
| Odyssey-Fex e UI-Odyssey-Fex | 3.8676 | 30 | 0.0005 |
| Confidence Intervals 95% | 2.8926 - 0.9365 | | |
| Mean of difference | 0.6129 | | |

Afterwards, in what refers to semantics, we obtained $t\text{-Test} = 3.1526$ and a $p\text{-value} = 0.0036$, as shown on Table VII, that is, the t value is higher than the tabulated 2.0423, and its $p\text{-value}$ is significantly lower than 0.05, from which we can **reject** the H_0 hypothesis and **accept** the H_1 hypothesis, concluding that the UI-Odyssey-Fex notation has adequate semantics to represent UI elements when compared with the Odyssey-Fex notation.

TABLE VII. PAIRED T-TEST QUESTION 3 - SEMANTIC

| Treatments | Paired t-Test | Degrees of Freedom | p-value |
|---------------------------------|-----------------|--------------------|---------|
| Odyssey-Fex e UI-Odyssey-Fex | 3.1526 | 30 | 0.0036 |
| Confidence Intervals 95% | 0.2044 - 0.9679 | | |
| Mean of difference | 0.5806 | | |

VI. FINAL CONSIDERATIONS

Requirements of a Software Product Line (SPL) are represented with Feature Models. There are several notations for this representation [6] [2] [15] [1] [3], which have distinct properties, though all have the means to represent mandatory and optional features of a system family within an SPL. Conversely, we have observed that some notations do not cover the representation of some non-functional aspects of an SPL, specially related to User Interface (UI) elements.

To overcome this limitation, this research has analyzed the representation of UI features in the building of Feature Models (FM) in SPL, comparing the notation Odyssey-Fex with its extension, UI-Odyssey-Fex. These notations were evaluated based on three Research Questions (RQ), each of which meant to evaluate advantages regarding ease-of-use, effort required and syntactic and semantic representativeness, respectively. The results show that regarding ease-of-use and required effort, there is no statistically significant difference between the two notations. However, there is a significant advantage to representativeness in the use of UI-Odyssey-Fex, when compared to Odyssey-Fex.

We have identified two major threats to the experiment. The first is the size of the sample, as well as its being comprised of a group that may not be representative of the population. The second is the fact that the experiment only compares two

notations, whereas ideally a comparison with other common notations, such as FODA, would be desirable. In future works, we hope to replicate this study with a different sample and more notations, in order to provide a more strong evidences about the benefits and drawbacks of representing UI elements using different FM notations.

REFERENCES

- [1] Q. Boucher, G. Perrouin, and P. Heymans. Deriving configuration interfaces from feature models: A vision paper. In *Proceedings of the Sixth International Workshop on Variability Modeling of Software-Intensive Systems*, pages 37–44, New York, NY, USA, 2012. ACM.
- [2] K. Czarnecki, S. Helsen, and U. Eisenecker. Staged configuration using feature models. In R. Nord, editor, *Software Product Lines*, volume 3154 of *Lecture Notes in Computer Science*, pages 266–283. Springer Berlin Heidelberg, 2004.
- [3] R. F. de Oliveira. Formalization and consistency checking in variabilities modeling. Master thesis, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, 2006.
- [4] J. Favaro and S. Mazzini. Extending featurseb with concepts from systems engineering. In S. Edwards and G. Kulczycki, editors, *Formal Foundations of Reuse and Domain Engineering*, volume 5791 of *Lecture Notes in Computer Science*, pages 41–50. Springer Berlin Heidelberg, 2009.
- [5] K. Kang, S. Cohen, J. Hess, W. Nowak, and S. Peterson. Feature-oriented domain analysis (FODA) feasibility study. Technical report, Software Engineering Institute, Carnegie Mellon University Pittsburgh, PA., 1990.
- [6] K. C. Kang, S. Kim, J. Lee, K. Kim, E. Shin, and M. Huh. Form: A feature-oriented reuse method with domain-specific reference architectures. *Ann. Softw. Eng.*, 5:143–168, Jan. 1998.
- [7] G. B. Klaus Pohl and F. van der Linden. *Software Product Lines Engineering: Foundations, Principles, and Techniques*. Springer - Verlag Berlin Heidelberg, KünkleLopka, Heindelberg, UE, 2010.
- [8] N. J. Miler. *A Engenharia de Aplicações no Contexto da Reutilização Baseada em Modelos de Domínio*. PhD thesis, Federal Univeristy do Rio de Janeiro, Rio de Janeiro, Brazil, 2000.
- [9] Q. L. Nguyen. Non-functional requirements analysis modeling for software product lines. In *Modeling in Software Engineering, 2009. MISE '09. ICSE Workshop on*, pages 56–61, May 2009.
- [10] L. of Software Engineering COPPE UFRJ. Odyssey project, Setembro 2014.
- [11] A. Pleuss, B. Hauptmann, D. Dhungana, and G. Botterweck. User interface engineering for software product lines: The dilemma between automation and usability. In *Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems, EICS '12*, pages 25–34, New York, NY, USA, 2012. ACM.
- [12] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015.
- [13] L. R. Soares, P. Potena, I. d. C. Machado, I. Crnkovic, and E. S. d. Almeida. Analysis of non-functional properties in software product lines: A systematic review. In *Proceedings of the 2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA '14*, pages 328–335, Washington, DC, USA, 2014. IEEE Computer Society.
- [14] J. Tukey. *Exploratory Data Analysis*. Addison-Wesley series in behavioral science. Addison-Wesley Publishing Company, 1977.
- [15] J. van Gurp, J. Bosch, and M. Svahnberg. On the notion of variability in software product lines. In *Proceedings. Working IEEE/IFIP Conference on Software Architecture, 2001*, pages 45–55, August 2001.
- [16] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering: An Introduction*. Springer - Verlag Berlin Heidelberg, Norwell, MA, USA, 2012.