Comprehending Feature Models Expressed in CVL

Iris Reinhartz-Berger[1], Kathrin Figl[2], and Øystein Haugen[3]

[1] Department of Information Systems, University of Haifa, Israel
iris@is.haifa.ac.il
[2] Institute for Information Systems & New Media, Vienna, Austria
kathrin.figl@wu.ac.at
[3] SINTEF and Department of Informatics, University of Oslo, Norway
oystein.haugen@sintef.no

Abstract. Feature modeling is a common way to present and manage variability of software and systems. As a prerequisite for effective variability management is comprehensible representation, the main aim of this paper is to investigate difficulties in understanding feature models. In particular, we focus on the comprehensibility of feature models as expressed in Common Variability Language (CVL), which was recommended for adoption as a standard by the Architectural Board of the Object Management Group. Using an experimental approach with participants familiar and unfamiliar with feature modeling, we analyzed comprehensibility in terms of comprehension score, time spent to complete tasks, and perceived difficulty of different feature modeling constructs. The results showed that familiarity with feature modeling did not influence the comprehension of mandatory, optional, and alternative features, although unfamiliar modelers perceived these elements more difficult than familiar modelers. OR relations were perceived as difficult regardless of the familiarity level, while constraints were significantly better understood by familiar modelers. The time spent to complete tasks was higher for familiar modelers.

Keywords: Variability analysis, Software Product Line Engineering, Model Comprehension.

1 Introduction

With the proliferation of software systems as an essential part of almost any business, their requirements increased and became more complex. Against this background, the variety of software artifacts has also heightened and a fundamental challenge is figuring out how to manage this variety. One way to tackle these challenges is analyzing and representing variability. Variability is extensively studied in the field of Software Product Line Engineering [1, 2] which aims at supporting the development and maintenance of families of software products, termed software product lines.

A systematic review published in 2011 [3] shows that variability management in software product lines is primarily done utilizing feature-oriented modeling. A *feature diagram* is a tree or graph that describes the end-user visible characteristics (features)

J. Dingel et al. (Eds.): MODELS 2014, LNCS 8767, pp. 501–517, 2014.

of systems in a software product line and the relationships and constraints (dependencies) between them. Different feature-oriented languages have been proposed over the years. However, no standard feature-oriented modeling language has emerged yet. Recently, some initiatives in this direction started in the Object Management Group (OMG), yielding the submission of the Common Variability Language (CVL) proposal [4]. CVL is a domain-independent language for specifying and resolving variability. It facilitates the specification and resolution of variability over any instance of a Meta-Object Facility (MOF)-based language (such an instance is termed a base model). Lately, CVL was recommended for adoption as a standard by the Architectural Board of OMG.

A prerequisite for effective variability management is comprehensible representation of variability. However, there are few empirical studies that analyze difficulties in understanding variability representations in general and feature diagrams in particular. To fill this gap, the main aim of this study was to examine the cognitive difficulty of understanding variability in feature modeling. In order to address this challenge we conducted an exploratory study using CVL models, perceiving CVL as an emerging standard that uses feature-oriented principles for specifying and representing variability. We concentrate on the variability abstraction part of CVL, which provides constructs for specifying variability without defining the concrete consequences on the base model. We further seek to answer in this study how modeler's familiarity with feature modeling influences comprehension difficulties. Differences between novices and experts can point on specific problems in comprehension. In addition, it is of specific interest how the target user group of CVL, modelers who are familiar with feature modeling, comprehend CVL and can easily switch from feature models to CVL models, as comprehensibility of a new modeling language is an important basis for its acceptance in practice.

The rest of the paper is structured as follows. Section 2 reviews the related work and provides the required background on CVL. Section 3 elaborates on the experiment design and procedure, while Section 4 presents the analysis procedure and the results. Section 5 discusses the results and the threats to validity. Finally, Section 6 summarizes and points on future research directions.

2 Related Work and Background

2.1 Comparison of Feature Modeling Languages

Comparison of feature modeling languages is mainly done using classification frameworks or according to lists of characteristics. Istoan et al. [5], for example, suggest a metamodel-based classification of variability modeling approaches: methods may use a single (unique) model to represent both commonality and variability or distinguish and keep separate the variability model from the base model. Methods that use a single model may annotate the base model by means of extensions or combine a general, reusable variability meta-model with different domain metamodels. Methods that use separate models specify the variability model using notations such as feature diagrams, decision models, or CVL [4]. Istoan et al. further compared the artifacts of these methods in the metamodel and model levels.

Czarnecki et al. [6] compared feature modeling and decision modeling along ten dimensions: applications, unit of variability (features vs. decisions), orthogonality, data types, hierarchy, dependencies and constraints, mapping to artifacts, binding time and mode, modularity, and tool aspects. They further showed how the main properties of feature modeling and decision modeling are reflected in three specific methods including an initial version of CVL.

Schobbens et al. [7] surveyed and compared seven feature diagram notations. These notations differ in their graph types (trees vs. directed acyclic graphs – DAG), the supported node types (e.g., cardinality support), the supported graphical constraint types (namely, "requires", "excludes", none, or both), and the supported textual constraint types (i.e., textual composition rules support). In a later work, Heymans et al. [8] evaluated the formal properties of feature diagram languages using Krogstie et al.'s semiotic quality framework [9] and Harel and Rumpe's guidelines for defining formal visual languages [10]. The list of evaluation criteria included: (1) expressiveness: what can the language express? (2) embeddability: can the structure of a diagram be kept when translated to another language? and (3) succinctness: how big are the expressions of one and the same semantic object?

Djebbi and Salinesi [11] provided a comparative survey on four feature diagram languages for requirements variability modeling. The languages are compared according to a list of criteria that includes readability, simplicity and expressiveness, type distinction, documentation, dependencies, evolution, adaptability, scalability, support, unification, and standardizeability.

Haugen et al. [12] proposed a reference model for comparing feature modeling approaches. This model makes distinction between the generic sphere, which includes feature models and product line models, and the specific sphere, which includes feature selection and product models. Standard languages, annotations, and special domain-specific languages are compared based on this reference model.

Comparing product line architecture design methods, Matinlassi [13] suggested an evaluation framework that is based on Normative Information Model-based Systems Analysis and Design (NIMSAD) [14]. According to this framework, there are four essential categories of elements for method evaluation: (1) context, including specific goals, product line aspects, application domains, and methods inputs/outputs; (2) user, including target groups, motivation, needed skills, and guidance; (3) contents, including method structure, artifacts, architectural viewpoints, language, variability, and tool support; and (4) validation, including method maturity and architecture quality.

All the above studies concentrate on the expressiveness of the compared languages, while usability-related issues follow a "feature comparison" approach. The drawback of such an approach is the subjectivity in developing both the comparison checklist and its interpretation. In addition, the above studies neglect comprehensibility, which is an important issue in modeling, as the abstract goal of modeling is to formally describe some aspects of the physical and social world around us for the purpose of understanding and communication [15].

Recent research has started to examine comprehensibility aspects of variability modeling languages. The work in [8] looks into comprehensibility appropriateness, namely whether or not language users understand all possible statements of the language. Comprehensibility appropriateness is, however, handled subjectively through embeddability and succinctness.

Conducting two experiments, Reinhartz-Berger and Tsoury [16, 17] compared the comprehensibility of Cardinality-Based Feature Modeling (CBFM) [18] and Application-based DOmain Modeling (ADOM) [19]. The comparison is based on comprehensibility of commonality- and variability-related concepts, including mandatory and optional elements, dependencies, variation points and variants. The experiments were conducted with small numbers of participants with similar background and experience. Accordingly, the conclusions were limited and concentrated on the differences between the methods: the feature-oriented CBFM and the UML-based ADOM.

To summarize, comprehensibility of feature modeling languages still needs to be assessed empirically. To fill this gap, we concentrate in this study on identifying difficulties in understanding feature models expressed in CVL. Next, we briefly present CVL.

2.2 Common Variability Lanaguge (CVL)

As noted, CVL facilitates the specification and resolution of variability over any base model defined by a metamodel. Its architecture consists of variability abstraction and variability realization. *Variability abstraction* supports modeling and resolving variability without referring to the exact nature of the variability with respect to the base model. *Variability realization*, on the other hand, supports modifying the base model during the process of transforming the base model into a product model.

In this study, which concentrates on the variability abstraction part of CVL, the main examined concepts are VSpecs (variability specifications), their relationships, and constraints. *VSpecs* are technically similar to features in feature modeling and can be divided into four subclasses: (1) choices – require yes/no decisions, (2) variables – require providing values of specified types, (3) VClassifier (variability classifiers) – require creating instances and resolving their variability, and (4) CVSpec – composite VSpecs. The VSpecs are organized in trees that represent logical constraints on their resolutions (configurations). VSpec children are related to their parents higher in the tree in two different ways: (1) Mandatory or optional: The positive resolution of a child may be determined by the resolution of the parent (mandatory) or can be independently determined (optional). (2) Group multiplicity: A range is given to specify how many total positive resolutions must be found among the children: XOR/alternative – exactly one, OR – at least one.

Finally, constraints express dependencies between elements of the variability model that go beyond what is captured in the tree structure. Two types of constraints are primarily supported in CVL: (1) A implies B – if A is selected, then B should be selected too (this constraint is known as “requires” in feature modeling), and (2) Not (A and B) – if A is selected, then B should not be selected and vice versa (this constraint is known as “excludes” in feature modeling).

As an example, consider Fig. 4(a) in the appendix. This figure describes the allowed variability in Skoda Yeti cars, a sport utility vehicle model of Skoda: the fuel of these cars is either diesel or benzin; the gear is either manual or automatic; the drive is either 2-weel-drive or 4x4; and the gadget level is either active or adventure. These configurations are further restricted by the constraints, e.g., a Skoda Yeti car whose gadget level is active and its fuel is diesel must have a manual gear (i.e., it cannot have an automatic gear). Fig. 4(b) further elaborates on the possible configurations of extra features in Skoda Yeti cars: such a car may have heated front pane, sunset, or panorama roof; it may have parking heater; and it may have styling package (and if so it may also have offroad styling).

3 Experiment Design and Procedure

3.1 Research Goal

Following Wohlin et al.'s guidelines [20], the goal of our exploratory study was to:

Analyze the effect of feature modeling familiarity on the comprehension difficulties of CVL, an emerging common variability language;

For the purpose of evaluation;

With respect to comprehension score in terms of percentage of correct solution, time spent to complete tasks, and participants' perception of difficulty;

From the point of view of modelers;

In the context of students of information systems, informatics and business.

3.2 Research Planning and Design

We derived from the above goal the following research questions and settings:

RQ1: Are there differences in the difficulty to comprehend CVL models for modelers who are familiar with feature modeling in general, in comparison to those who are unfamiliar with it?

To answer this research question we followed a quasi-experimental between-subject design. We classified participants into two distinct groups: participants familiar/unfamiliar with feature modeling. The exact classification procedure is explained in Section 3.6. The *independent variable* in this case was familiarity with feature modeling, while the *dependent variables* were comprehension scores (measured using the percentage of correct solution), time spent to complete tasks, and difficulty perception using self-rated difficulty of specific element types.

There is a long tradition of researching domain familiarity in terms of expert-novice differences in the area of system development. Petre [21] has argued that "experts 'see' differently and use different strategies than novice graphical programmers". Prior research has revealed that experts develop language-independent, abstract problem representations, so-called schemas, which make similar tasks for them easier. In light of these theoretical considerations we conjecture that comprehending

CVL models should be easier for modelers who are familiar with feature modeling. We hypothesize that as follows:

H1a. Familiarity with feature modeling will be positively associated with the comprehension score.

H1b. Familiarity with feature modeling will be positively associated with the time spent to complete tasks.

H1c. Familiarity with feature modeling will be negatively associated with the perceived difficulty of specific feature modeling constructs.

The second research question is divided into the following two parts:

RQ2a: Which feature modeling constructs are difficult to understand?
RQ2b: Are there specific difficulties in comprehending CVL models for modelers who are familiar with feature modeling, in comparison to those who are unfamiliar with it?

To answer the second research question we used a within-subjects design. The selected *independent variable* was element type, namely mandatory, optional, and alternative (XOR) features, as well as OR relations and feature constraints (dependencies). The *dependent variables* were comprehension score, time spent, and self-rated difficulty of the specific feature modeling elements. As it is not possible to construct comprehension tasks that only demand the understanding of "basic" elements, namely, mandatory, optional, and alternative features, we could only assess their subjective, but not their objective, difficulty. However, it was possible to objectively determine comprehension score and time to complete tasks related to OR relations and constraints.

We refrain from hypothesis development for the second research question, as it would not be helpful in such an exploratory setting [22]. While a considerable amount of literature has been published on expert-novice difference in general (RQ1), no research has been conducted on the comprehension of specific feature modeling constructs (RQ2) up to now. To our knowledge this is the first study evaluating comprehension of CVL, thus there is neither an adequate empirical nor a theoretical basis available to formulate a priori hypotheses.

3.3 Experimental Material

The *objects* of the experiment were two CVL models describing different sets of features of Skoda Yeti cars. The CVL models for the experiment were built by Prof. Haugen, who is the founder of CVL and is familiar with the possible Skoda Yeti configurations from the Norwegian Skoda public web pages. The first model (see Fig. 4a in the appendix), named "basic", describes basic features of Skoda Yeti. It includes mandatory and alternative features (using XOR relations), as well as four "implies" ("requires") constraints. The number of valid configurations is 6. The second model (see Fig. 4b in the appendix), named "expanded", describes advanced (extra) features. It includes optional features and an OR relationship, as well as two "excludes" constraints (phrased as "not (A and B)"). The number of valid configurations in this case is much higher.

3.4 Tasks

The tasks were embedded within an online questionnaire. On each model ten questions were asked, examining whether specific configurations of Skoda Yeti are allowed according to the model. These questions can be described as surface-level tasks which measure comprehension of models more directly than deep-level tasks, which require participants to work with the models in a usage context [23]. For our research goal of checking the comprehensibility of a relatively new language, surface-level model comprehension tasks are most appropriate.

The participants were presented with the model and one question at a time. They had to choose between the following answers: Correct, Wrong, Cannot be answered from model, I don't know. After answering a question, the participant proceeded to the next question, but could not return to previous questions. This way we could measure the time needed to answer an individual question. The questions used in the experiment appear in the appendix.

3.5 Procedure

The participants were requested to open the online questionnaire. There, they had to fill first a pre-questionnaire that was composed of two parts. The purpose of the first part of the pre-questionnaire was to obtain general information about the participants and their background, including age, gender, degree and subject of studies, and familiarity with feature modeling. To measure self-rated familiarity with feature modeling, which was one of the independent variables, we adopted the three-item modeling grammar familiarity scale of Recker [24]: (1) Overall, I am very familiar with feature diagrams; (2) I feel very confident in understanding feature diagrams; (3) I feel very competent in modeling feature diagrams.

The second part of the pre-questionnaire aimed to objectively examine the prior knowledge of the participants in modeling. This part included three models in three languages: ER, BPMN, and feature diagrams. The participants had to state in which language each model was specified and answer three comprehension questions about the models. Each question presented a statement and four possible answers: Correct, Wrong, Cannot be answered from model, I don't know.

After filling the pre-questionnaire, the participants were presented with slides explaining and exemplifying the variability abstraction part of CVL (the participants also got hard-copies of these slides which they could consult while answering the questions). The participants had to study CVL variability abstraction part on their own from the slides and proceed to the main questionnaire. The main questionnaire included two CVL models of Skoda Yeti cars and two sets of questions, each of which referred to a different model (basic vs. expanded). To avoid any order effects due to fading attention, we used two different samplings of the main questionnaire, in which models were arranged in a different sequence. As noted, the models and questions of the main questionnaire appear in the appendix.

The time spent on each question was recorded by the online questionnaire. No rigid time constraints were imposed on the participants.

After completing the questions on each model, the participants had to fill a post-part questionnaire that collected feedback on the tasks and the difficulty to understand different model elements (mandatory and optional features, XOR and OR relations). The answering options ranged from 1=very easy to 7=very difficult. In addition, the participants could report on difficulties they experienced in open text fields.

3.6 Participants

Participants were recruited from three different classes from information systems, informatics and business curricula with prior training in modeling. Indeed, it can be argued that using students to evaluate a language that aimed at software professionals is problematic, but it has been shown in [25] that students have a good understanding of the way industry behaves, and may work well as subjects in empirical studies in specific areas, such as requirements engineering. Additionally, students are a relatively homogenous group concerning knowledge about and experience with conceptual modeling [26].

The executions of the experiment took place in the spring semester of 2013 in three courses that deal with modeling in the University of Haifa, the Vienna University of Economics and Business, and the University of Oslo. To assure sufficient motivation, the experiment was defined as an obligatory exercise or participants received approximately 5% course credit for participating. A total of 38 students participated in the study.

We now turn to the categorization of participants as familiar/unfamiliar with feature modeling. Since CVL is a relatively new language there are so far no "experts", however familiarity with feature modeling can serve as an adequate proxy. We use an extreme group selection approach to define two clearly distinct groups on the continuous variable familiarity in order to heighten power of statistical tests [27]. 15 participants were categorized as not familiar with feature modeling. They had answered the question "are you familiar with feature diagrams?" negatively. Regarding those who had answered this question affirmatively, we crosschecked the mean score of the scale familiarity with feature diagrams (3 items). We also assessed the reliability of this scale, which was adequate (Cronbach's alpha = 0.96). According to Nunnally and Bernstein [28] Cronbach's alpha should be higher than 0.8 to combine items in a mean value. The answering options for this scale ranged from 1=strongly disagree to 7=strongly agree. We concluded that only participants with a mean threshold value of 3.5 for this scale could be regarded as "familiar". Based on this criterion, 6 participants were excluded. In addition we checked whether all of these "familiar" participants also had correctly identified the figure of the feature diagram in the pre-questionnaire. This objective test led to further exclusion of one participant. Overall 7 participants were excluded as they were neither unfamiliar nor sufficiently familiar with feature modeling, resulting in 16 participants who were categorized as familiar with feature modeling. The final sample consisted of 31 participants: 22 males (58%) and 16 females (42%) with a mean age of 28 years.

Table 1. Differences in participants' skills

	Unfamiliar with feature modeling (n=15)		Familiar with feature modeling (n=16)		Statistical Test
	Mean	SD	Mean	SD	
Feature modeling test score (3 points possible)	1.20	0.77	2.13	0.89	$T_{df=29}=-3.09$ $p=0.00$
Modeling score test (8 points possible)	4.00	1.25	4.19	0.91	$T_{df=29}=-0.48$ $p=0.64$

We performed t-tests to demonstrate differences between the groups (see Table 1 for detailed results). As expected, participants of the familiar group scored better in the feature modeling test, while no significant differences in the modeling pre-test were found. This means that the students in the unfamiliar group were familiar with modeling in general, but unfamiliar with feature modeling and these differences may affect their difficulties in comprehending CVL models.

4 Analysis Procedure and Experiment Results

Data analysis was performed using SPSS 19. We first present the effects of familiarity with feature modeling on comprehension. We then refer to difficulties to comprehend specific model elements. Finally, we present results on comprehension difficulties as perceived and reported by the participants.

4.1 Comprehension and Familiarity with Feature Modeling

To answer the first research question (RQ1: Are there differences in the difficulty to comprehend CVL models for modelers who are familiar with feature modeling in general, in comparison to those who are unfamiliar with it?), we performed an analysis of covariance (ANCOVA) for repeated measures (basic and expanded models) for each dependent variable (comprehension score and time spent). The independent variable was familiarity with feature modeling (familiar vs. unfamiliar). Familiarity had a significant influence on the comprehension score (F=2.60, p=0.01). Participants familiar with feature modeling achieved a comprehension score of 85% on average, while unfamiliar participants achieved only 69%, lending support to H1a, which had predicted a positive association between familiarity and comprehension score. In addition, the interaction effect of familiarity and model (basic vs. expanded) was significant (F=9.68, p=0.00). As can be seen in Fig. 1 there was almost no performance difference between familiar and unfamiliar modelers in the expanded model, but a clear difference in the basic model. As noted, the basic model included only XOR relations and mandatory features, as well as "implies" ("requires") constraints, while the expanded model included an OR relation and "excludes" constraints. H1a can only be partially supported in case of the basic model.

There was also a significant influence of familiarity on time (F=5.39, p=0.03). Contrary to our expectations, however, participants familiar with feature modeling took more time to complete tasks (418 seconds vs. 298 seconds). Thus, H1b could not be supported, i.e., familiarity is negatively associated with time spent.

To control for possible order effects, we had included a second independent variable in the analyses – model order – which represents the order in which models were presented to the participants. Model order had two values: ‘basic first’ – the basic model was presented first and ‘basic later’ – the expanded model was presented first. We found that the model order did not have any significant effect on the comprehension score, but it influenced the time spent; there was a significant interaction effect of model (basic vs. expanded) and model order (F=9.00, p=0.01). Participants spent more time on answering questions on the first model they were presented with than for the second model.

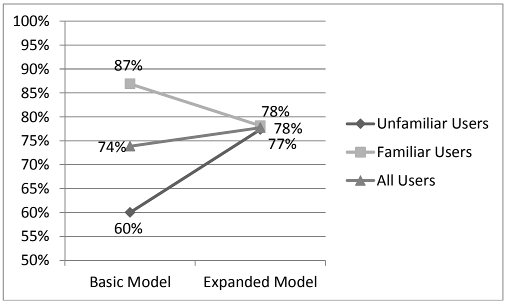

Fig. 1. Familiarity and Comprehension of Basic and Expanded Models

4.2 Difficulties to Comprehend Specific Model Elements

In order to answer the second research question (Which feature modeling constructs are difficult to understand? Are there specific difficulties in comprehending CVL for modelers who are familiar with feature modeling, in comparison to those who are unfamiliar with it?), we identified two model elements which could be addressed in seperation from other elements via comprehension tasks and whose comprehension deserve special attention: constraints and OR relations.

To check whether comprehension tasks which demand understanding of constraints in general are more difficult to understand than tasks without constraints, we performed again ANCOVAS. Overall, there was no difference between the comprehension of questions which involved constraints (79% correctness) and questions which did not involve constraints (81%). However, participants spent more time to solve questions without constraints (43 seconds) than with constraints (25 seconds; F=26.20, p=0.00). Familiarity has a significant influence on comprehension score (F=5.29, p=0.03) and time spent (F=4.74, p=0.04) as already reported in the prior analysis. We further observed that there is a significant interaction effect of familiarity and the existence of constraints (F=4.76, p=0.04) on comprehension score. As can be seen in Fig. 2 below, there is a larger performance difference of comprehension for questions involving constraints. While participants familiar with feature modeling achieved an average comprehension score of 90% on questions involving constraints, unfamiliar participants achieved only 67%. The difference in questions not involving constraints is smaller (84% vs. 78%, respectively). These results strengthen support for H1a (familiarity is positively associated with the comprehension score) in case constraints are involved.

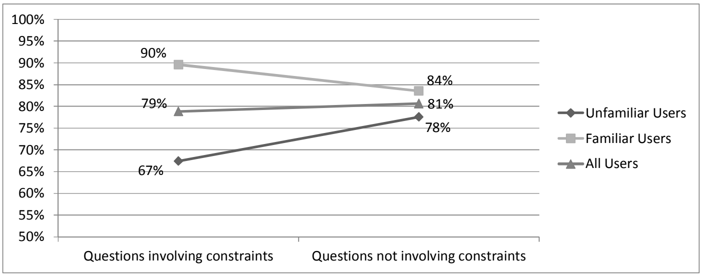

Fig. 2. Familiarity and Comprehension of Constraints

Another worth-mentioning result refers to OR relations, which were only included in the expanded model. We grouped questions according to the semantics of the OR relation: none – no option appears in the question, one – exactly one option appears, and many – two or more options appear. Comparisons between the three groups were analyzed with ANCOVAS. We found significant main effects of OR relations on comprehension score (F=3.18, p=0.05) and time spent (F=8.53, p=0.00). As can be seen in Fig. 3, questions including no option ('none') were the easiest for both groups. Questions including no option ('none') took on average most time (40 seconds), followed by one option selection ('one', 48 seconds) and multiple selections ('many', 26 seconds). The analysis further showed that for those tasks from the expanded model familiarity had no effect on the comprehension score, but on time, as discussed in the previous section. There were no significant interaction effects with familiarity.

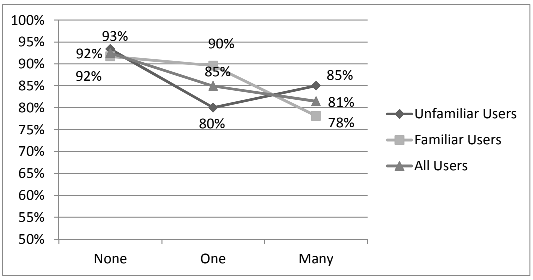

Fig. 3. Familiarity and Comprehension of OR Relations

4.3 Perceived Comprehension Difficulties

To find out whether familiarity with feature modeling changed the difficulties participants experienced, we performed a series of t-tests. Table 2 gives descriptive statistics and the details of the analyses. Note, we used 7-point scale for these items, with

1=very easy and 7=very difficult. Basic elements, namely, mandatory, optional, and alternative (XOR) features, were rated significantly more difficult by participants having no prior experience with feature models. There was no difference between the groups for the difficulty rating of OR relations, which were only included in the expanded model. As familiarity was associated negatively with perceived difficulty of specific feature modeling constructs in 4 of 5 cases (the gray rows in the table), the results lend support to H1c.

Regarding constraints, participants mentioned difficulties in the open text fields. Participants of the unfamiliar group mentioned that "learning and mapping constraints were a little [bit] difficult". Participants of the familiar group mentioned that "the 'implies' relationships were fairly difficult to understand" and "the NOT (A and B) [constraint] was difficult without [an] example" (although an example of such a constraint was included in the CVL slides). One participant of the familiar group also stated that "the constraints helped understand the model's intent very clearly". The participants further mentioned difficulties that we had not explicitly asked for, including the unclear meaning of parent-child relations in the feature tree ("It's hard to understand if feature can be used without his parent") and the lack of explicitness of all feature combinations ("Many combinations were not specified directly in the model").

Table 2. Comprehension difficulties as reported by the participants

	Unfamiliar with feature modeling		Familiar with feature modeling		Total		t	p
	M	SD	M	SD	M	SD		
mandatory features (basic model)	3.40	1.55	2.13	0.72	2.74	1.34	2.91	0.01
optional features (basic model)	3.67	1.63	2.19	0.83	2.90	1.47	3.14	0.00
optional features (expanded model)	3.33	1.35	2.31	1.01	2.81	1.28	2.40	0.02
XOR relations (basic model)	3.27	1.67	2.13	0.72	2.68	1.38	2.45	0.02
OR relations (expanded model)	3.07	1.39	2.56	1.09	2.81	1.25	1.13	0.27

5 Discussion and Threats to Validity

This study set out with the aim of assessing the comprehensibility of feature models expressed in CVL and identifying differences in comprehension difficulty for modelers familiar or unfamiliar with feature modeling principles. It was hypothesized that participants familiar with feature modeling would comprehend CVL models better and faster and experience less difficulties. And indeed, the results showed that comprehension score was higher and perceived difficulty lower for familiar modelers. This result is in line with findings of [29] who showed that familiarity with models in a specific domain also enables modelers to understand a new modeling language in that domain faster and with less effort.

It is interesting to note that the performance difference for familiar modelers was higher for the model which mainly included XOR relations than the second CVL model which included an OR relation. A possible explanation for this result is that OR relations in general are difficult to understand and that familiarity with feature modeling not necessarily trains modelers in understanding OR relations nor enables a cognitive advantage for understanding this construct. Further findings support this interpretation, because familiar and unfamiliar modelers subjectively rated the difficulty of OR relations similarly and while multiple selections ('many') were more difficult than 'one' or 'none' OR selections, familiarity did not interact with the comprehension of OR relations. Research findings on deductive reasoning with natural language connectives provide a theoretical explanation for the high cognitive difficulty of inclusive ORs. "OR" is likely to be misinterpreted in its exclusive form, not as an inclusive OR-operator [30]. Based on empirically detected comprehension difficulties, other model domains as process modeling have advised in modeling guidelines to avoid inclusive OR gateways altogether [31]. While this option is not feasibility for the area of feature modeling, still our results can be used to adopt training material and to inform modeling practitioners to be cautious in case of OR relations.

In addition, familiar modelers had a clear advantage in understanding textual constraints in CVL models in comparison to unfamiliar modelers. The observed difference could be attributed to the familiarity of the modelers with feature modeling. An implication of this result for modeling practice includes the need to put a specific emphasis on making constraints easier to understand for novice modelers. A possibility to achieve this goal might be to place textual constraints spatially close to respective features in the model, thus following the spatial contiguity rule [32] if possible. Indeed, the currently developed CVL tool supports associating constraints closely to the relevant features.

Unexpectedly, familiarity with feature modeling was negatively related to time. This result might sound counterintuitive at first sight, however there are possible interpretations. Knowing the notation may increase the doubts, requiring more time and not necessarily improving the performance. Participants might also be more motivated to solve the comprehension tasks correctly and work harder and longer to solve them if they already have prior experience in that modeling domain. Furthermore, unfamiliar modelers have lower comprehension of OR relations and constraints than familiar modelers, but familiar modelers spent long time to achieve their better comprehension. This may be due to the perception of familiar modelers that OR relations and constraints are more difficult – a perception that made them spend more time on those questions. This is similar to experienced drivers who slow down properly in curves.

As with all studies, the reader should bear in mind that a number of limitations associated with laboratory experiments need to be acknowledged [20]. One source of weakness regarding external validity includes the use of student subjects. However, the participants had received training in modeling and, therefore, we do believe that they serve as an adequate proxy for future modelers of CVL. In addition, we assured that random influences to the experimental setting were low to improve conclusion validity. First, participants were committed to the experiment by making the experiment an obligatory exercise or giving course credit (of about 5%) for participation. Second, the students self-studied CVL, so no influence of the lecturers' capabilities, knowledge, and opinions were introduced to the CVL training.

Despite the clear support for the hypothesized associations, the generalizability of findings reported here should be undertaken with caution, because we could only include two different CVL models in the study and we selected a specific feature modeling language – the variability abstraction part of CVL. As the two models included in the questionnaire were typical representatives we argue that they provided a reasonable test of comprehensibility, thus, assuring construct validity. The selection of the language was done perceiving CVL as an emerging standard which systematically includes the main feature modeling concepts. However, only further experiments with other feature modeling languages and models can confirm or disconfirm the generalizability of our results.

6 Conclusions and Future Work

The purpose of the current study was to determine comprehensibility of feature models as expressed in CVL and possible difficulties for different user groups. We found that familiarity with feature modeling did not influence the comprehension of basic elements, namely mandatory, optional, and alternative features, although unfamiliar modelers perceived these elements more difficult than familiar modelers. OR relations were perceived as difficult regardless of the familiarity level, while constraints were significantly better understood by familiar modelers. The time spent to complete tasks was higher for familiar modelers. The findings from this study add to the current body of knowledge on feature modeling by investigating comprehension and have relevant implications for practice and research. The results further add to a growing body of literature on novice-expert differences in modeling research. Our results also offer important suggestions for training in feature modeling. While understanding of mandatory and optional features, constraints and XOR relations is important in the training of users new to feature modeling, understanding of OR relations remains difficult even for experienced modelers. In addition, results can be used to guide feature modeling language developers in their design efforts and help in ongoing revision of CVL. Our initial results provide evidence that CVL is comprehensible for novice and expert modelers and that they can read such models after a short training.

Several opportunities for future research emerge from our study. For instance, further experimental investigations with a larger variety of models would be required to estimate comprehension difficulty of specific modeling elements. Future studies could also extend this work and examine difficulties in modeling (and not just understanding) variability using CVL. Finally, comprehension difficulties can be researched in additional feature modeling languages, as well as in the variability realization part of CVL.

References

1. Pohl, K., Böckle, G., van der Linden, F.: Software Product Line Engineering: Foundations, Principles, and Techniques. Springer (2005)
2. Clements, P., Northrop, L.: Software Product Lines: Practices and Patterns. Addison-Wesley, Boston (2001)
3. Chen, L., Ali Babar, M.: A systematic review of evaluation of variability management approaches in software product lines. Information and Software Technology 53, 344–362 (2011)
4. Haugen, Ø.: Common Variability Language (CVL) – OMG Revised Submission. OMG document ad/2012-08-05 (2012)
5. Istoan, P., Klein, J., Perouin, G., Jezequel, J.-M.: A Metamodel-based Classification of Variability Modeling Approaches. In: VARiability for You Workshop, pp. 23–32 (2011)
6. Czarnecki, K., Grünbacher, P., Rabiser, R., Schmid, K., Wąsowski, A.: Cool features and tough decisions: A comparison of variability modeling approaches. In: Proceedings of the Sixth International Workshop on Variability Modeling of Software-Intensive Systems, pp. 173–182. ACM, Leipzig (2012)
7. Schobbens, P.-Y., Heymans, P., Trigaux, J.-C.: Feature Diagrams: A Survey and a Formal Semantics. In: Proceedings of the 14th IEEE International Requirements Engineering Conference, pp. 136-145. IEEE Computer Society (2006)
8. Heymans, P., Schobbens, P.Y., Trigaux, J.C., Bontemps, Y., Matulevicius, R., Classen, A.: Evaluating formal properties of feature diagram languages. IET Software 2, 281–302 (2008)
9. Krogstie, J., Sindre, G., Jørgensen, H.D.: Process Models Representing Knowledge for Action: A Revised Quality Framework. European Journal of Information Systems 15, 91–102 (2006)
10. Harel, D., Rumpe, B.: Meaningful Modeling: What's the Semantics of "Semantics"? Computer 37, 64–72 (2004)
11. Djebbi, O., Salinesi, C.: Criteria for Comparing Requirements Variability Modeling Notations for Product Lines. In: Workshops on Comparative Evaluation in Requirements Engineering, pp. 20–35 (2006)
12. Haugen, Ø., Møller-Pedersen, B., Oldevik, J.: Comparison of System Family Modeling Approaches. In: Obbink, H., Pohl, K. (eds.) SPLC 2005. LNCS, vol. 3714, pp. 102–112. Springer, Heidelberg (2005)
13. Matinlassi, M.: Comparison of software product line architecture design methods: COPA, FAST, FORM, KobrA and QADA. In: Proceedings of the 26th International Conference on Software Engineering, ICSE 2004, pp. 127–136 (2004)
14. Jayaratna, N.: Understanding and Evaluating Methodologies: NIMSAD, a Systematic Framework. McGraw-Hill, Inc. (1994)
15. Mylopoulos, J.: Conceptual Modeling and Telos. In: Loucopoulos, P., Zicari, R. (eds.) Conceptual Modeling, pp. 49–68. John Wiley and Sons, New York (1992)
16. Reinhartz-Berger, I., Tsoury, A.: Experimenting with the Comprehension of Feature-Oriented and UML-Based Core Assets. In: Halpin, T., Nurcan, S., Krogstie, J., Soffer, P., Proper, E., Schmidt, R., Bider, I. (eds.) BPMDS 2011 and EMMSAD 2011. LNBIP, vol. 81, pp. 468–482. Springer, Heidelberg (2011)
17. Reinhartz-Berger, I., Tsoury, A.: Specification and Utilization of Core Assets: Feature-Oriented vs. UML-Based Methods. In: De Troyer, O., Bauzer Medeiros, C., Billen, R., Hallot, P., Simitsis, A., Van Mingroot, H. (eds.) ER Workshops 2011. LNCS, vol. 6999, pp. 302–311. Springer, Heidelberg (2011)

18. Czarnecki, K., Kim, C.H.P.: Cardinality-based feature modeling and constraints: a progress report. In: International Workshop on Software Factories at OOPSLA. ACM (2005)
19. Reinhartz-Berger, I., Sturm, A.: Utilizing domain models for application design and validation. Inf. Softw. Technol. 51, 1275–1289 (2009)
20. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M., Regnell, B., Wesslén, A.: Experimentation in Software Engineering – An Introduction. Kluwer Academic Publishers (2000)
21. Petre, M.: Why looking isn't always seeing: readership skills and graphical programming. Commun. ACM 38, 33–44 (1995)
22. Kumar, S., Karoli, V.: Handbook of Business Research Methods. Thakur Publishers (2011)
23. Parsons, J., Cole, L.: What do the Pictures mean? Guidelines for Experimental Evaluation of Representation Fidelity in Diagrammatical Conceptual Modeling Techniques. Data and Knowledge Engineering 55 (2005)
24. Recker, J.: Continued Use of Process Modeling Grammars: The Impact of Individual Difference Factors. European Journal of Information Systems 19, 76–92 (2010)
25. Svahnberg, M., Aurum, A., Wohlin, C.: Using students as subjects - an empirical evaluation. In: Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, pp. 288–290. ACM, Kaiserslautern (2008)
26. Siau, K., Loo, P.-P.: Identifying Difficulties in Learning UML. Information Systems Management 23, 43–51 (2006)
27. Preacher, K., Rucker, D., MacCallum, R., Nicewander, W.: Use of the Extreme Groups Approach: A Critical Reexamination and New Recommendations. Psychol Methods 10, 178–192 (2005)
28. Nunnally, J.C., Bernstein, I.H.: Psychometric Theory. McGraw-Hill, New York (1994)
29. Recker, J., Dreiling, A.: The Effects of Content Presentation Format and User Characteristics on Novice Developers' Understanding of Process Models. Communications of the Association for Information Systems 22 (2011)
30. Naess, A.: A Study of 'Or'. Synthese 13, 49–60 (1961)
31. Mendling, J., Reijers, H., van der Aalst, W.M.P.: Seven process modeling guidelines (7PMG). Information and Software Technology 52 (2010)
32. Moody, D.L.: The "Physics" of Notations: Towards a Scientific Basis for Constructing Visual Notations in Software Engineering. IEEE Transactions on Software Engineering 35, 756–779 (2009)

Appendix: Experiment's Models and Questions

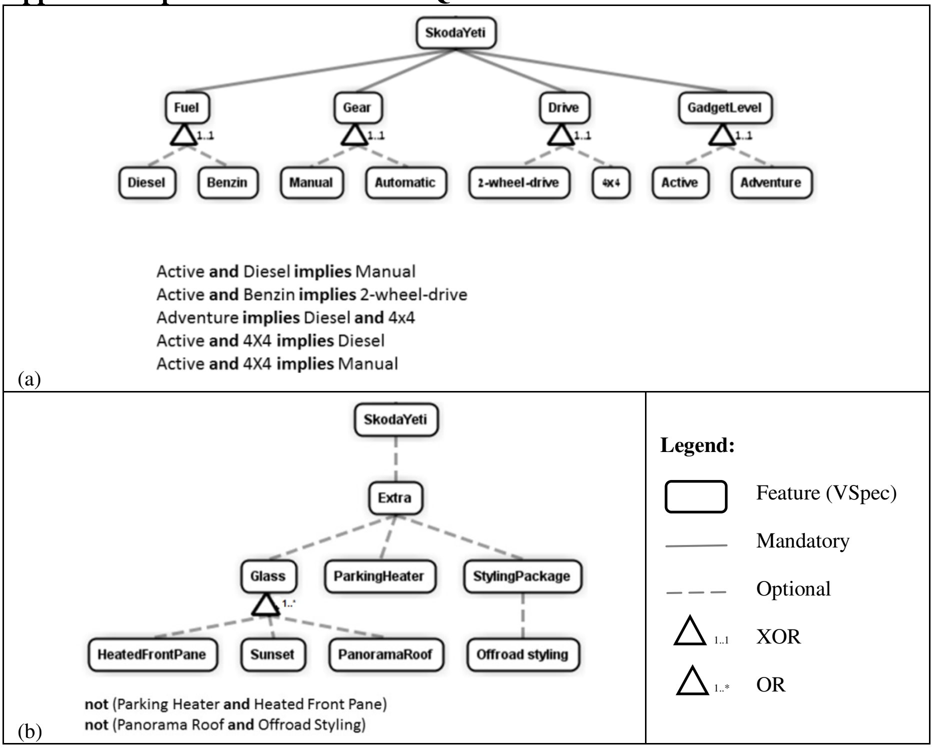

Fig. 4. The CVL models used in the experiment: (a) the basic model and (b) the expanded model

Basic Model Questions	Expanded Model Questions
A Skoda Yeti car can have the following combination of features:	
1. Manual and Diesel	1. Parking-Heater and Styling-Package
2. Adventure and Benzin	2. Panorama-Roof and Offroad-Styling
3. Automatic and 4x4	3. Parking-Heater and Offroad-Styling
4. Adventure and 2-wheel-drive	4. Parking-Heater and Heated-Front-Pane
5. Active and Diesel and Automatic	5. Parking-Heater and Styling-Package and Offroad-Styling
6. Diesel and Automatic and 4x4	6. Sunset and Parking-Heater and Styling-Package
7. Active and Benzin and 4x4	7. Heated-Front-Pane and Sunset and Panorama-Roof
8. Adventure and Manual and 4x4	8. Sunset and Panorama-Roof and Parking-Heater and Offroad-Styling
9. Active and Benzin and Manual and 2-wheel-drive	9. Heated-Front-Pane and Sunset and Styling-Package and Offroad-Styling
10. Automatic and Adventure and Benzin and 2-wheel-drive	10. Heated-Front-Pane and Sunset and Panorama-Roof and Styling-Package