ACM Copyright Notice

Published in: ***Proceedings of the Seventh International Workshop on Variability Modelling of Software-intensive Systems (VaMoS'13),*** January 2013

"A Survey of Variability Modeling in Industrial Practice"

Cite as:

Thorsten Berger, Ralf Rublack, Divya Nair, Joanne M. Atlee, Martin Becker, Krzysztof Czarnecki, and Andrzej Wąsowski. 2013. A survey of variability modeling in industrial practice. In Proceedings of the Seventh International Workshop on Variability Modelling of Software-intensive Systems (VaMoS '13). ACM, New York, NY, USA, , Article 7 , 8 pages.

BibTex:

```
@inproceedings{Berger:2013:SVM:2430502.2430513,
 author = {Berger, Thorsten and Rublack, Ralf and Nair, Divya and Atlee, Joanne M. and Becker, Martin and Czarnecki, Krzysztof and W\k{a}sowski, Andrzej},
 title = {A Survey of Variability Modeling in Industrial Practice},
 booktitle = {Proceedings of the Seventh International Workshop on Variability Modelling of Software-intensive Systems},
 series = {VaMoS '13},
 year = {2013},
 pages = {7:1--7:8},}
```

DOI: http://dx.doi.org/10.1145/2430502.2430513

A Survey of Variability Modeling in Industrial Practice

Thorsten Berger
IT University of Copenhagen,
University of Leipzig
thbe@itu.dk

Ralf Rublack
University of Leipzig
ralf@rublack.org

Divya Nair
University of Waterloo, ON
dknair@cs.uwaterloo.ca

Joanne M. Atlee
University of Waterloo, ON
jmatlee@uwaterloo.ca

Martin Becker
Fraunhofer IESE
beckerm@iese.fhg.de

Krzysztof Czarnecki
University of Waterloo, ON
kczarnec@gsd.uwaterloo.ca

Andrzej Wąsowski
IT University of Copenhagen
wasowski@itu.dk

ABSTRACT

Over more than two decades, numerous variability modeling techniques have been introduced in academia and industry. However, little is known about the actual use of these techniques. While dozens of experience reports on software product line engineering exist, only very few focus on variability modeling. This lack of empirical data threatens the validity of existing techniques, and hinders their improvement. As part of our effort to improve empirical understanding of variability modeling, we present the results of a survey questionnaire distributed to industrial practitioners. These results provide insights into application scenarios and perceived benefits of variability modeling, the notations and tools used, the scale of industrial models, and experienced challenges and mitigation strategies.

1. INTRODUCTION

Variability modeling is the discipline of explicitly representing variability in dedicated models that describe the common and variable characteristics of products in a software product line. It is core to many software product line engineering (SPLE) methodologies used in research [?, ?, ?] and industry [?, ?], and has received increasing attention over the last twenty years. Numerous techniques have been proposed in academia and industry, with a large number of different notations and tools available.

Surprisingly, there are few empirical studies that aim at understanding the use of variability modeling in practice [?, ?]. Although many experience reports on SPLE exist, only a minority focus on variability modeling. This lack of empirical data i) threatens the applicability of academic approaches in industry, and ii) hinders the improvement of industrial techniques. It is important to understand actual usage scenarios, including the applied notations and tools, as well as perceived benefits and challenges.

We attempt to address this gap with a survey on industrial variability modeling. This survey is part of our larger ongoing effort to improve the empirical understanding of variability modeling. We have previously investigated the systems software domain [3], including very large and highly configurable systems, such as the Linux kernel. We are currently conducting case studies using semi-structured interviews. In this paper, we focus on the results of a survey questionnaire distributed to industrial practitioners. Our questionnaire was guided by the following research questions:

RQ1: What variability modeling notations and tools are used?
RQ2: What are the scales of industrial models?
RQ3: What are perceived benefits and challenges of variability modeling?

The questionnaire also aimed at setting variability modeling into context by identifying i) the domain and adoption strategies used in the respective product line projects, ii) the artifacts whose variability is described by models, and iii) the roles and experience of our participants. Questionnaires are among the few research tools that can be used to obtain an empirical record of industrial practices. Comprehensive artifact studies—as we conducted for open source systems software—are often infeasible due to highly protected variability models, which usually contain core strategic knowledge of a company. Although our quantitative results need to be complemented by in-depth qualitative case studies, we believe they provide practitioners and researchers with a good overview of the variety of variability modeling practices applied in industry.

We proceed as follows. We first motivate our objectives with previous work in Section 2, followed by the design and distribution criteria of our questionnaire in Section 3. We report results in Section 4, covering contextual information, notations and tools, model scales, and perceived benefits and challenges. We discuss threats to validity in Section 5, summarize related work in Section 6, and conclude in Section 7.

2. MOTIVATION AND OBJECTIVES

We previously studied concepts of open source variability modeling languages and characteristics of their models from the systems software domain [3, 4, 17, 2, 16]. The

VaMoS '13 Pisa, Italy

languages CDL and Kconfig, stemming from the embedded operating system eCos and the Linux kernel, share concepts with feature and decision modeling [7], but have additional concepts that help to scale variability modeling, such as visibility, modularization, or derived features and defaults. We hypothesize that languages developed in industry have similar concepts. In this survey, our objective is to identify applied notations and tools.

The primary application of the CDL and Kconfig languages and their tools is the configuration of systems. However, variability models can be used for a multitude of applications, such as planning and scoping (domain modeling) of product lines, helping developers in maintaining a summary of available features, or supporting marketing. We aim to identify usage scenarios and perceived values of variability modeling.

The variability models we identified in the open source systems software domain have sizes up to 6320 features. While experience reports indicate the existence of very large industrial models [15, 18], we strive to identify a larger sample of models and elicit their characteristics.

A challenge for Linux developers is dependency management within the variability model [?, ?]. Significant effort is spent on fixing dependencies to ensure valid derived systems. Visualization is another challenge [3], since all investigated models are very wide and shallow. In this light, our survey aims to confirm known and identify new challenges in industrial practice.

Strategies to combat experienced challenges can be manifold. Our study of the systems software domain revealed, among others, the use of modularization and encapsulation concepts, and automated reasoning (e.g. an inference engine to resolve configuration conflicts). We expect to find similar strategies in industrial practice.

3. METHODOLOGY

As this survey is embedded into our currently ongoing framework study on industrial variability modeling, we first describe the high-level methodology, and then expand on the methodology of the survey reported in this paper—that is, the design, distribution, and analysis of the questionnaire.

The goal of our framework project is to qualitatively and quantitatively understand characteristics of industrial variability models, their creation process, and the languages and tools used. Therefore, we follow a mixed-methods approach: first, we design and distribute a broadly scoped survey questionnaire; second, we analyze its results and identify case studies for deeper analysis; third, we perform interviews; and fourth, analyze results using grounded theory with open coding [10]. The questionnaire's primary purpose is not to test hypotheses, but to provide an overview on industrial practices and to identify interesting targets of case studies.

Questionnaire design. The main design criteria of our questionnaire was to keep it simple and short. Its target group comprises practitioners that participated in industrial SPLE efforts and applied variability modeling techniques. We formulated 15 questionnaire questions according to our main research questions. They first elicit the attitudes towards variability modeling and perceived benefits, then the notations and tools applied by our participants, then the scales of models, followed by experienced challenges and applied mitigation strategies. At the end, we formulated contextual questions, including personal information. The latter questions aim to verify results, identify duplicate product

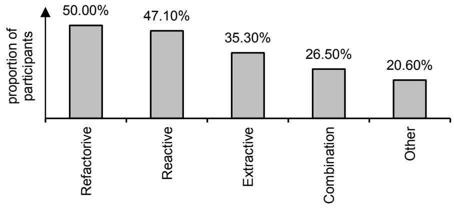

Figure 1: Product line adoption strategies

lines, and to allow for follow-ups with clarification questions or interview invitations. Most questions were multiple-select, with the possibility to give additional open-text answers. The final questionnaire is available via our project website.[1]

Questionnaire distribution. We implemented the questionnaire using the online tool SurveyGizmo[2], and distributed it to over 60 practitioners and researchers with industrial experience on SPLE *and* variability modeling. Our initial list of invites comprised our own industrial partners, academic colleagues with industrial background, partners of Fraunhofer IESE, and companies from the software product line Hall of Fame[3]. Respondents were invited to forward the questionnaire to further colleagues. We also spread the survey questionnaire at the VaMoS'12 workshop.

Questionnaire analysis. Before analysis, to ensure that responses reflect actual industrial practice, we filter out responses that omit personal contact data, and those that selected only the researcher role in the corresponding question. Our main analysis tools are diagrams and manual aggregations of responses to individual questions. We use a narrative style to report results, along with careful interpretations. Due to the relatively small dataset (42 responses), it is not feasible to apply automated rule learning techniques to obtain statistically significant correlations. We also cannot calculate correlation coefficients due to the multiple-select nature of most questions. However, as interesting conjectures showed up during analysis, we created cross tabulations (pivot tables) between individual options of the multiple-select questions.

4. RESULTS

We received 42 responses from participants originating from 16 different countries, primarily Germany (24%), Canada (12%), USA (12%), Sweden (7%), Austria (5%), Brazil (5%), Norway (5%) and Spain (5%). Single responses came from China, Denmark, France, Greece, India, Poland, Switzerland, and the United Kingdom. As mentioned above, we filtered out five responses that came from researchers, and two that did not contain personal data (so background of the respondent could not be established). The results below stem from the remaining 35 responses. Among these are some responses from researchers, or former researchers, who also indicate industrial affiliation.

Most of our respondents (57%) have more than five years of experience with SPLE; half of them report even more than ten years. Very few (8%) report professional experience of less than one year. The most frequent role is modeler (71%), followed by researcher (68%), developer (51%), and team leader (40%). We only had one marketing expert. Finally, 17% of respondents mentioned architect, 8% consultant, 3%

[1] gsd.uwaterloo.ca/industrial-variability-modeling
[2] www.surveygizmo.com
[3] www.splc.net/fame.html

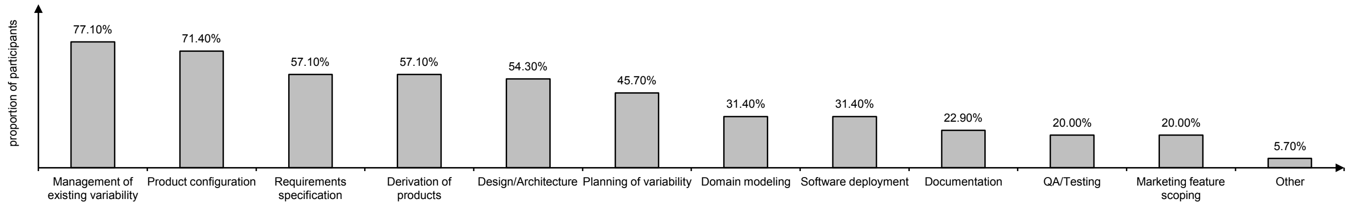

Figure 2: Perceived values of variability modeling

admin, and 3% project coordinator as an open-text answer.

Interestingly, researchers involved in industrial SPLE predominantly participate in modeling (71% of the cases) and development (in 50% of the cases) activities. In fact, for some participants, we know that they played both an academic and industrial role, provided consulting, or switched between industry and academia over time.

4.1 Context of Variability Modeling

To set variability modeling into context, we asked for three characteristics of the respective product line projects: the application domain, the adoption strategy, and the artifacts whose variability is modeled. All potentially influence practices, perceived benefits, and challenges.

Our respondents reported a wide range of application domains of their product line projects. Table 1 summarizes our classification of all the responses to this open question, and the number of occurrences of each domain category. Most of our respondents apply variability modeling for automotive, industrial, and enterprise software. However, many quite diverse domains are also reported. The category "other" aggregates domains reported only once, including underwater acoustics systems, geographical information systems, travel, and logistics.

We asked for the product line adoption strategies that were applied by our respondents. Following Krueger [11], we distinguish between *pro-active* (product line is developed before any product is derived), *re-active* (a single product is evolved into a product line), and *extractive* (existing products are re-engineered into a product line) strategies.

As summarized in Fig. **??**, only 35% of our participants reported using a pro-active strategy for any of their product lines. As many as 50% applied an extractive strategy, and 47% a re-active one. The numbers do not add to 100%, as each respondend might have participated in development of more than one SPL. A quarter of the respondents reported using a combination of these three strategies. Although the pro-active strategy with its upfront scoping and platform development processes is regarded as the typical SPLE approach, our finding supports the alternative hypothesis common in the community: that only a minority of industrial product lines are planned pro-actively. It seems necessary to refocus SPLE research on strategies, methodologies and tools for re-engineering existing systems into product lines.

Table 1: Software product line domains

application domain	count
automotive	11
industrial applications and energy	8
enterprise and eCommerce	7
aerospace and defense	5
medical	4
consumer electronics	2
government	2
telecommunication	2
other	10

Most of our respondents' variability models represent the variability of software components (72%) and of the source code, which represents static variability (64%). In contrast, dynamic variability at runtime is only mentioned by 36%. Interestingly, 53% of our respondents reported requirements and the architecture/design; followed by the platform with 39%. Slightly minor is variability in test cases (25%), libraries (14%), and documentation (17%). Among the open-text answers were build files, DSL instances, roadmaps, calibration files, specification models and knowledge representation, and even roadmaps and release plans.

4.2 Benefit of Variability Modeling

In our earlier studies on systems software, we concluded that the dominating application of variability modeling is supporting product configuration. Somewhat in contrast to this observation, the industrial product line engineers indicate a much wider range of benefits of variability modeling, not just configuring products. See Fig. 1.

Respondents indicate that the management of existing variability as the main benefit of variability modeling, followed by product configuration, requirements specification, derivation of products, design, and architecture. Interestingly, seven respondents see benefits in feature scoping for marketing purposes. In open-text responses, the respondents complemented our provided answers with maintenance and cost estimations for new features, and planning of development and evolution.

Consistent with our expectations, all participants who use variability modeling for planning variability use feature-based representation. All but one of these use feature models (the only exception relies on *"Boolean variables for features, Boolean expressions for constraints, no grouping or clustering"*). This indicates that the coarser and more abstract notion of features (as opposed to lower-level decisions or variation points) indeed facilitates planning.

4.3 Notations and Tools

We asked participants whether they use separate models to describe variability, or whether existing implementation artifacts are annotated: 72% selected the former, 47% the latter, and 31% both options. The open-text responses included statements such as separate DSLs, delta modeling, and annotations provided by the Spring framework.

Notations. As summarized in Fig. 5, feature models are the most frequently reported notation; followed by a mix of formal techniques, such as UML-based notations, DSLs, architecture description languages, and non-formal ap-

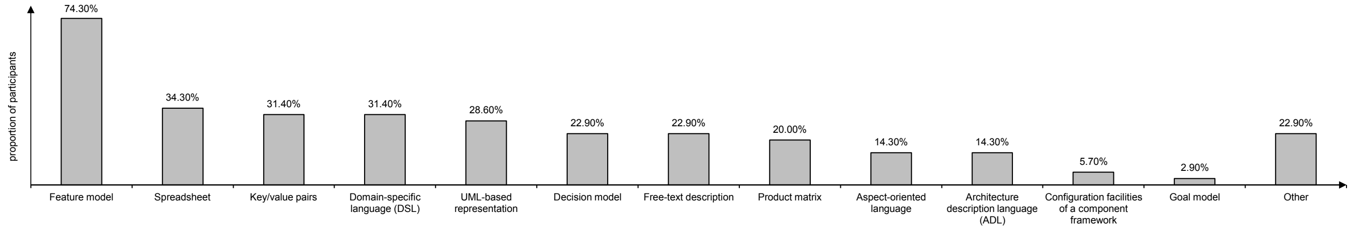

Figure 3: Notations used to specify variability

proaches, such as spreadsheets, free-text descriptions, and flat key/value pairs in XML- or text-based property files. Two participants reported using the configuration facilities of a component framework to represent variability. The open-text responses revealed even more notations, including Design Structure Matrix [?] and CVL [6], a recent proposal for OMG's upcoming Common Variability Language standard.

Most respondents have used more than one notation among their product line projects. The average number of notations selected is three. Very few participants (8%) chose only one notation. On the other hand, a maximum of eight different notations (feature model, decision model, DSL, product matrix, spreadsheet, facilities of a component framework, key/value pairs in property files, and free-text descriptions) was used by one participant within product lines from the medical, industry automation, and energy domain.

These results confirm that a variety of notations is used in industry. Since most of our participants relied on several notations, this finding threatens variability management approaches that focus on exactly one variability representation. We speculate that industrial product line engineers use different techniques to satisfy domain-specific needs. This speculation is supported by our earlier studies of variability modeling languages in open source systems, which all contained domain specific, or even project-specific language constructs. So far, we have no strong empirical evidence that one-size-fits-all solutions are applicable in industry without specific adaptations.

Tools. Finally, we asked about the use of variability modeling tools. Our multiple-select question listed 13 tools drawn from the literature and our project partners. All except Oracle's Siebel configurator were one or more respondents. Fig. 6 shows the results.

The pure::variants tool from pure::systems was the most frequently selected one (35% responses), followed by Gears from BigLever Software (23%). This gives no indication of market penetration, but is an unsurprising consequence of geographic distribution of our respondents (65% are located in Europe). All other tools play only a minor role in the participating projects—although we know that many of these have significant market shares, such as the SAP configurator. Each of these tools, except the Siebel configurator, was only reported once or twice.

As open-text answers, many respondents reported i) home-grown domain-specific tools, ii) other open source, and iii) other commercial tools. These responses reveal many smaller tools of which we were not aware so far; as listed in Table **??**.

As many as 38% of respondents have used a home-grown domain-specific tool, for example based on Eclipse EMF or IBM Rational Software Architect, or as extensions to the MontiArc architecture description language and toolset, and to Simulink. One respondent mentions home-grown tool support for XML-based representations, text tables, and source code comments.

Slightly less than a third of the participants have used an open source tool, such as a CVL prototype, the CVM framework [?], Hephaestus [5], and SPREBA [19]. Although these were conceived in the research community—unlike pure::variants and Gears—the survey confirms their application in industry. 27% of our respondents used a different commercial tool, such as the decision-model-based tool Tecnalia PLUM [1], v.control [14], SparxSystems Enterprise Architect, or simply Microsoft Excel.

In summary, the identified variety of tools is surprising. While literature mostly cites the commercial tools pure::variants and Gears, which are also dominant among our participants, many smaller academic and industrial solutions are also applied. This finding shows that industry has developed home-grown solutions that are unknown to researchers. This situation reinforces our observations from the open source studies, where especially eCos boasts a well-engineered language with complex reasoning support, developed completely independently of the research community. Whether similar solutions were developed among our industrial participants remains an interesting question.

4.4 Scales and Constraints of Models

Two questions aimed at identifying the scales of our participants' variability models. First, we asked for the number of models of specific size ranges—carefully formulated as the number of "units of variability", since we anticipated heterogeneous representations of variability. Second, we asked respondents to specify these units, for instance, as features, decisions, variation points or calibration parameters.

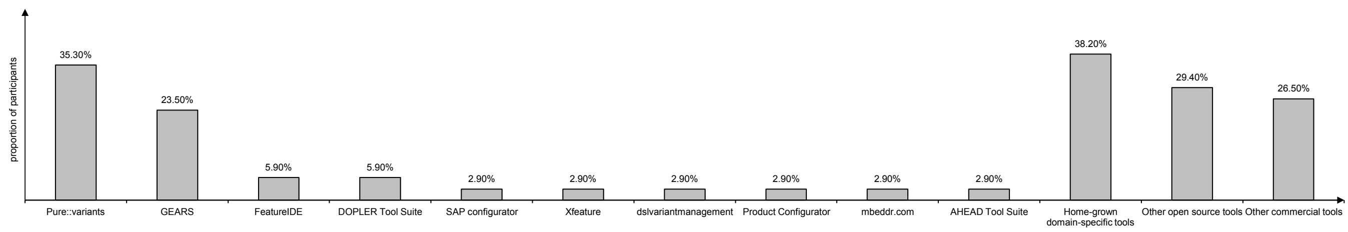

Figure 4: Tools used to model variability

Table 2: Tools reported as open-text answers

	open-text answer	count
open source	CVL/CVLTool	2
	SPREBA (ADORA)	2
	Eclipse EMF/Ecore	2
	CVM	1
	Hephaestus tool to manage variabilities in SPLs	1
	Spring	1
	Xtext	1
commercial	Enterprise Architect	3
	PLUM	2
	CWAdvisor/ MS Excel	1
	IBM RSA	1
	Microsoft Dynamics AX	1
	systemweaver	1
	v.control	1
home-grown domain-specific	BCS, Dialog	1
	Delta-MontiArc (Architectural Variability), Delta-Simulink (Variability of SL models)	1
	Eclipse-based graphical toolset: FMT	1
	Extension of IBM RSA to model variability in UML	1
	Internally developed tools	1
	MasterCraft	1
	Other internally developed modeling and generative tools	1
	ParameterManager	1
	Various XML, text-tables, source code comments	1
	different tools for different types of variability	1

Table 2 summarizes the proportion of reported scales: the columns reflect size ranges, the rows the number of models, and the cells the proportion of our participants who declared to have a certain number of models in the specific size range.

Although two thirds of our respondents reported very small models with fewer than 50 features, the results show that very large models exist. 26% reported models with more than 10,000 units, where most of them stem from the automotive, defense, and further industrial application domains. Finally, three quarters of our participants reported experience with more than just one model.

In most cases (80%), the units of variability were features, and variation points (73%); followed by *configuration options* (70%), *decisions* (33%), and *calibration parameters* (27%). As an open-text response, one participant reported *deltas*. As calibration parameters stem from the automotive industry, it is not surprising that their usecorrelates with the automotive domain (except underwater acoustics/sonar and defense, which also use calibration parameters). On average, 2–3 units were reported. All participants with large models over 10,000 units reported using features or decisions. This finding confirms the existence of such large feature or decision models, in addition to experience reports. Further confirmation with in-depth case studies is part of our ongoing work.

Table 3: Scales of variability models

	<50[1]	51–100[1]	101–1000[1]	1001–10000[1]	>10000[1]
0 models	8.6%	22.9%	22.9%	42.9%	40.0%
1 model	40.0%	20.0%	28.6%	14.3%	14.3%
2–5 models	11.4%	14.3%	8.6%	0.0%	5.7%
>5 models	17.1%	8.6%	11.4%	8.6%	5.7%
sum (≥1 model)	68.5%	42.9%	38.6%	22.9%	25.7%

[1] units (e.g. features, decisions, variation points)

Finally, we asked whether explicit dependencies, such as requires or excludes, are explicitly modeled. Such dependencies are usually referred to as cross-tree constraints in feature models, since they cross-cut hierarchy dependencies. They are known to critically influence reasoners [**?**].

Among our participants, 80% confirmed the existence of explicitly modeled dependencies, whereas 45% report an average of fewer than 25% of units having dependencies; 21% between 26–50%, 7% between 51–75%, and 21% between 76–100%. Although a further interpretation of these results requires artifact studies, they indicate that the proportions of dependencies in industrial models are lower than in our previously analyzed systems software models.

4.5 Challenges and Mitigation Strategies

We asked about specific complexity issues that our practitioners faced with variability modeling. As shown in Fig. 8, the most frequently reported problem is variability model evolution, followed by the visualization of models. Dependency management and problems with the configuration process, such as resolving conflicts, have only slightly lower frequencies. The average number of challenges reported was 2–3, with 11% of respondents selecting all five options, and only three respondents selecting none.

Finally, the open-text answers to the question about challenges included, among others, modularization for multi product lines [12], testing, model reduction, but also statements such as *"getting developers to understand why we do this, and the correct patterns to use"*.

The strategies that our respondents use to tackle these issues are manifold, as shown in Fig. 9. 66% of respondents organize multiple models in a hierarchy; however, since this proportion is surprisingly high, we suspect that some accidentally referred to the intra-model organization (e.g., feature hierarchy). Two other frequent strategies—decomposition into multiple models and the use of model reasoners—confirm observations from our previous studies.

The open-text answers reveal further interesting opinions. One confirms the observation from our previous studies that variability models are fragile and need to be controlled centrally by a small team. The participant recommends to *"assign configuration / variability dependent tasks to a small selection of people"*. Other responses emphasize the removal of unnecessary variability, or *"much manual work, rule engines for consistency checking and value propagation (no SAT solvers)"*.

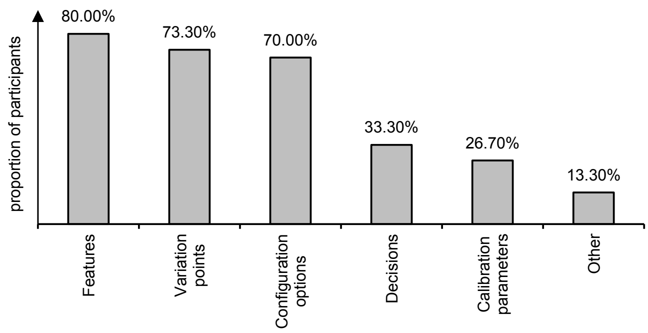

Figure 5: Units of variability

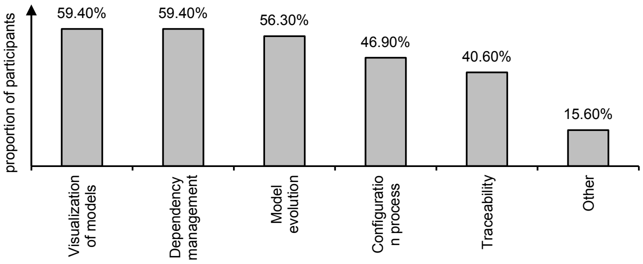

Figure 6: Reported complexity problems

4.6 Further Observations

A comment on the questionnaire itself discusses the applicability and necessity of variability modeling in non-embedded and non-system-software with modern dynamic languages. A former researcher in variability modeling, who is now in industry with more than 10 years of experience, stated:

> *"Both the field and this study could use a broadening of perspective. My day job is building Java based server side software, which tends to be one of a kind, non product line type software. Java is a rich language and technologies such as spring and maven provide very rich variability tooling. Additionally, using things like puppet for deploying into cloud architectures as well as staging and testing infrastructure means that we have a lot of variability. We deploy in different configurations to different data centers, use feature flags as well as AB testing to test new functionality, etc. My feeling is that the research field still assumes a traditional low tech embedded software perspective where the lack of a lot of things need to be compensated for with variability modeling tooling and cumbersome build systems. So, I don't model variability, instead I make software that has variation points that are explicitly configurable. The activities of developing and designing when following a continuous deployment model are inseparable."*

This statement is interesting from two perspectives. First, there exists research in dynamic product lines, but variability modeling for dynamic settings is immature. Second, since many tools and technologies are used in practice to realize dynamic variability, we believe that the actual challenge lies in establishing an unified overview of the variability, particularly when multiple realization technologies are used. Such an overview is a prerequisite to manage the variability effectively. For example, Schwanninger et al. [?] researched variability modeling for dynamic and heterogeneous product lines at Siemens VAI. This research challenge will likely become more prevalent with the success of cloud solutions [?].

5. THREATS TO VALIDITY

All of our conclusions are based on declared information, voluntarily given by participants in an online questionnaire. As such they are suspect to usual threats: the information given is subject to interpretation of the sources, and might reflect their beliefs, rather than actual practice/facts. We have mitigated this threat to an extent by requiring that authors are not anonymous and filtering answers not coming from industrial participants. We know that all answers come from trusted sources. We intend to follow up on the information gathered with qualitative interviews to obtain a more detailed, complementary picture.

A threat to the internal validity is that our practitioners misunderstood questions due to inconsistent terminology that exists in the variability modeling field. We mitigated this threat by test-driving the questionnaire with a range of colleagues during its development. We also provided open-text fields for comments.

We approached only practitioners who apply variability modeling. To elicit their attitude towards variability modeling, we used a Likert scale to ask whether they find variability modeling useful. Since most found it either definitely useful (55%) or useful (35%), and only two had a neutral and one a negative opinion, we assume that most participants were successful with variability modeling. We cannot conclude much about experiences of organizations that failed in similar processes.

To quantify success and failure of variability modeling in industry, a study on a wider range of SPLE projects is necessary, which is out of our scope. This quantification, in combination with a qualitative analysis of affecting factors, would constitute interesting future work.

6. RELATED WORK

We now discuss related experience reports, empirical studies, and literature reviews on variability modeling.

A significant number of experience reports from successful industrial SPLE adoptions is provided by van der Linden et al. [?], the software product line community's "Hall of Fame", and the Software Engineering Institute's catalog of case studies [?]. While most of these provide valuable insights into economic, organizational, and process aspects of the respective projects, only a few report on variability modeling, and with few details given. Exceptions are the reports of Jepsen and Beuche [?], Grünbacher et al. [?], Riebisch et al. [?], Reiser et al. [?], Gillan et al. [?]. Unfortunately, neither the models, nor detailed data on them is available.

Empirical studies on variability management in general comprise the work of Chen et al. [?], who elicit perceived

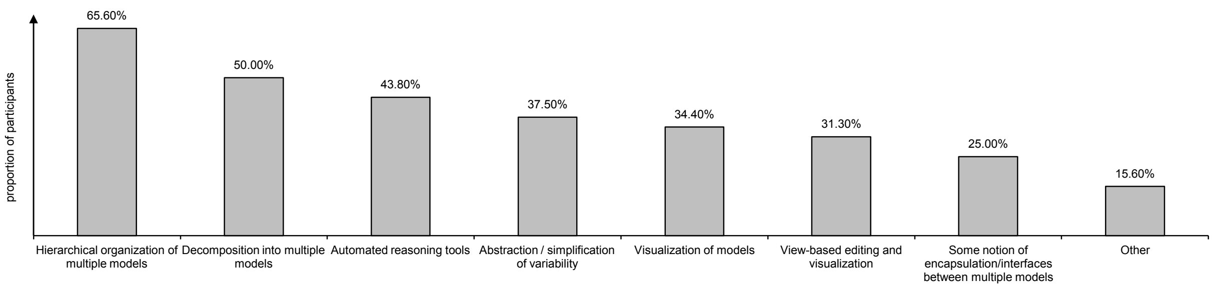

Figure 7: Reported strategies to combat complexity problems

challenges from eleven industrial practitioners using focus group research. Although their focus is broader than ours, they confirm some of the challenges we identified, such as the visualization and evolution of models—the latter particularly with respect to dependency management. Thörn et al. [?] survey variability management practices among regional Swedish small and medium-scale companies. Similar to our findings, they identify a broad range of variability documentation approaches, from ad-hoc to sophisticated approaches. Furthermore, they observe a correlation between the awareness for systematic variability management to the company size. Their work complements ours; however, our work is more focused on variability modeling practices and technology; furthermore, we also cover international and large-scale organizations.

A tool survey by Djebbi et al. [?] studies four variability modeling tools in order to compare their capabilities with 34 expectations from industry. The latter comprise modeling requirements, such as feature modeling concepts (e.g., a feature hierarchy, mandatory and optional features, cross-tree constraints). Unfortunately, the source of these expectations remains unknown. Rabiser et al.'s survey on requirements for product derivation [?] ranks interactive support for resolving variability highest, requiring scalable model reasoners. Furthermore, Hubaux et al.'s survey questionnaire on configuration challenges in Linux and eCos [?] identifies a lack of guidance and the low quality of advice for making choices in the configurators. Our work is an extension to this line of research, providing additional insights into variability modeling practices in industry.

Finally, literature reviews on variability management support the motivation of our work—the lack of empirical studies on variability modeling. Hubaux et al. [?] review literature on the application of feature modeling in practice. They conclude that only two percent of the reviewed papers discuss practical experiences. Furthermore, Chen et al. [?, ?] conduct two literature reviews that emphasize the lack of thorough empirical evaluation, and of "detailed comparative analyses to show the relative advantages or disadvantages of different variability management approaches".

7. CONCLUSIONS

We have presented a survey of variability modeling practices in industrial software product lines. The survey has been based on 35 answers to an online questionnaire coming from industrial actors all over the world.

Industrial SPL developers indicate a much wider applicability of SPLs than we know from earlier studies of open source practice, exceeding simple configuration modeling with variability management, requirements specification, design and architecture planning.

We have observed huge diversity in how detailed and large the models are. Most participants use models with fewer than 50 variability units, but about a quarter declare models exceeding 10,000 units. This indicates that variability models have different use cases in different organizations, and that very different requirements need to be satisfied by tools to address these.

We observed very high heterogeneity of notations and tools. Feature models and tools based on feature modeling are clearly dominating. However, high diversity in the remaining part of the stage indicates that no golden standard is set yet. The industry is still experimenting with various solutions. We hypothesize that, similarly to the open source systems software, this diversity can be attributed to the need to address domain specific aspects in handling variability. If this is the case, it is unlikely that we will ever see uniformization of the variability notation and tooling.

An average participant used three different notations in her projects, thus it would be valuable to research tools and methods that support using a diversity of notations. Interestingly, separate variability models are used more often than annotative approaches. This is a good indicator for success of the upcoming OMG standard, the Common Variability Language CVL [6], which addresses the need for separate variability modeling.

A significant majority of the respondents follow re-active and extractive strategies for introducing product lines. This contrasts with focus in SPLE research, which is tipped towards pro-active strategies, starting with domain analysis and architectural design. The community might need to refocus SPLE research towards methods and tools supporting development of product lines from existing legacy software assets, such as re-engineering and reverse-engineering.

Product line engineers face challenges of technical and organizational nature. Technical challenges include primarily evolution and visualization of models, followed by dependency management. Process-oriented challenges center around ensuring support in the organization. Also, it seems necessary to limit access to the central and fragile variability model to a very small team.

In the next step of this project we intend to analyze our current set of eight conducted qualitative interviews, aiming at more with some of our respondents. The intention is to obtain a deep insight into practices and challenges. One, but not the only, question worth investigation is: what variability management, the reported primary benefit of variability modeling, indeed means in these organizations and why do they find it most beneficial. We believe that answering question this with proper empirical data is essential for the research community working on product lines and variability.

8. REFERENCES

[1] A. Aldazabal and S. Erofeev. Product line unified modeler (plum). 2007.

[2] T. Berger and S. She. Formal semantics of the CDL language, 2010. Technical Note. Available at `http://thorsten-berger.net/cdl_semantics.pdf`.

[3] T. Berger, S. She, R. Lotufo, A. Wasowski, and K. Czarnecki. Variability modeling in the systems software domain. Technical Report GSDLAB-TR 2012-07-06, Generative Software Development Laboratory, University of Waterloo, 2012. Available at `http://gsd.uwaterloo.ca/tr/vm-2012-berger`.

[4] T. Berger, S. She, R. Lotufo, A. Wąsowski, and K. Czarnecki. Variability modeling in the real: A perspective from the operating systems domain. In *Proceedings of the 25th IEEE/ACM International Conference on Automated Software Engineering*, ASE'10, 2010.

[5] R. Bonifácio, L. Teixeira, and P. Borba. Hephaestus: A tool for managing spl variabilities. 2009.

[6] CVL Submission Team. Common variability language (cvl), omg revised submission, 2012. Available at `http://www.omgwiki.org/variability/lib/exe/`

fetch.php?id=start&cache=cache&media=cvl-revised-submission.pdf.

[7] K. Czarnecki, P. Grünbacher, R. Rabiser, K. Schmid, and A. Wasowski. 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*, VAMOS'12, 2012.

[8] S. H. Khandkar. Open coding. Lecture material, available at http://pages.cpsc.ucalgary.ca/~saul/wiki/uploads/CPSC681/open-coding.pdf, 2009.

[9] C. W. Krueger. Easing the transition to software mass customization. In *4th International Workshop on Software Product-Family Engineering*, PFE'01, 2001.

[10] C. W. Krueger. New methods in software product line development. In *Proceedings of the 10th International on Software Product Line Conference*, SPLC'06, 2006.

[11] R. Likert. A technique for the measurement of attitudes. *Archives of psychology*, 1932.

[12] R. Lotufo, S. She, T. Berger, K. Czarnecki, and A. Wasowski. Evolution of the Linux kernel variability model. In *Proceedings of the 14th International Conference on Software Product Lines: Going Beyond*, SPLC'10, 2010.

[13] S. Mann and G. Rock. Dealing with variability in architecture de-scriptions to support automotive product lines. In *Proceedings of the Third International Workshop on Variability Modelling of Software-Intensive Systems*, VaMoS'09, 2009.

[14] J. Refstrup. Adapting to change: Architecture, processes and tools: A closer look at hp's experience in evolving the owen software product line. In *Proceedings of the 13th International Software Product Line Conference*, SPLC'09, 2009. Keynote, available at http://www.sei.cmu.edu/splc2009/files/SPLC2009AdoptingtoChange_Owen_2009_final.pdf.

[15] S. She and T. Berger. Formal semantics of the Kconfig language, 2010. Technical Note. Available at http://eng.uwaterloo.ca/~shshe/kconfig_semantics.pdf.

[16] S. She, R. Lotufo, T. Berger, A. Wasowski, and K. Czarnecki. The variability model of the Linux kernel. In *Proceedings of the Fourth International Workshop on Variability Modelling of Software-Intensive Systems*, VaMoS'10, 2010.

[17] M. Steger, C. Tischer, B. Boss, A. Müller, O. Pertler, W. Stolz, and S. Ferber. Introducing pla at bosch gasoline systems: Experiences and practices. In *Proceedings of the Third International Software Product Line Conference*, SPLC'04, 2004.

[18] R. Stoiber and M. Glinz. Modeling and managing tacit product line requirements knowledge. In *Proceedings of the 2009 Second International Workshop on Managing Requirements Knowledge*, MARK'09, 2009.