A Preliminary Review on the Application of Feature Diagrams in Practice

Arnaud Hubaux, Andreas Classen*
PReCISE Research Centre
University of Namur, Belgium
Email: {ahu, acs}@info.fundp.ac.be

Marcílio Mendonça
University of Waterloo, Canada
Email: marcilio@csg.uwaterloo.ca

Patrick Heymans
PReCISE Research Centre
University of Namur, Belgium
Email: phe@info.fundp.ac.be

***Abstract*—For two decades, feature diagrams have been intensively studied as a means to specify variability and pilot configuration in software product line engineering. Surprisingly though, it seems that very few reports on the use of feature diagrams in practice are available. To test this claim, we started a systematic review of such reports. In the collected material, we tried to identify positive and negative feedback on the use of feature diagrams. In this paper, we present the first results of this work in progress and discuss the opportunity of extending it to a fully systematic review on a wider scale.**

I. Introduction

Software product line (SPL) engineering (SPLE) has long been promoted as a cost-effective means to build customisable products out of reusable assets [1]. SPL development is traditionally a two-activity process [2]. The first activity, called *domain engineering*, consists in developing a set of reusable assets that can be configured and combined to create different products of the SPL. A key point in domain engineering is the identification and documentation of variability of the SPL. Typically, the variability is documented in a variability model. Variability models can take different forms and shapes, and address artifacts of different natures [2]. Here we focus on a particular kind of model, namely *feature diagrams* (FDs) [3], which are typically used to provide a technology-independent and high-level representation of variability. The second activity is *application engineering* during which variability is progressively resolved: the stakeholders decide which features from the FD are selected for inclusion in the final product and which are discarded until the product is completely *configured* [4].

The research community has worked intensively on FDs for more than two decades. This resulted notably in regular enhancements of their expressiveness (e.g. [4]), formalisation (e.g. [5]) and automated reasoning tools (e.g. [6]). However, thorough studies of the fitness of FDs wrt. industrial problems are hard to find. By fitness we mean the ability to fulfil a particular function or meet a particular need. Our discussions with researchers and practitioners also support this "informal observation". In this context, we believe it is necessary to clear up the matter and provide evidence of the presence or absence of supportive material. This concern can be translated into the following research question: *What evidence do we have of the fitness of FDs in practice?*

*FNRS Research Fellow.

In order to answer the research question, one possibility is to conduct a systematic literature review [7]. However, the time and resources needed to perform such a review are considerable. It is then often recommended to start with a smaller scale review that serves as an opportunity and feasibility study for a full systematic review [8]. In this paper, we follow this recommendation and propose a preliminary review based on a limited sample of paper sources.

Another systematic review in the field of variability modelling was published recently by Chen *et al.* [9] but our objectives are different:

- we *focus on FDs* and not on variability models in general;
- we inventorise *applications* of FDs in practice rather than approaches to model variability.

The main contribution of this paper is preliminary evidence that industrial application of FDs has been barely tackled in the literature and that there is an opportunity to conduct a full systematic review. In the long run, the expected benefits from our study are the following:

- a comprehensive inventory of success stories as well as failures of FDs in industry that can be used to define guidelines to help practitioners decide whether FDs are appropriate or not to their specific needs;
- repeated updates of our study will allow to assess the progress of the acceptance of FDs in industry;
- the collected observations can be used to identify the major research problems and define a practice-driven research roadmap for FD modelling and analysis.

The paper is structured as follows. Section II presents the review method and Section III describes the results of our analysis. Section IV discusses the limitations of our results and opportunities for conducting a full systematic review.

II. Review method

Since this study is only meant to be a pilot for a possible full systematic review, we relaxed some of the guidelines proposed by Kitchenham [7]. We thus refer to this pilot as a semi-systematic review. We will elaborate more on this in Section IV.

The initial paper base consists of the proceedings of all editions of (1) the *software product line conference* (SPLC) and co-located workshops for the years 2000, 2002, and

2004–2009, (2) the workshop on *product family engineering* (PFE) for the years 2001 and 2003, and (3) the workshop on *Variability Modelling of Software-intensive Systems* (VaMoS) for the years 2007–2009. PFE has been retained to ensure the continuity of publications between 2000 and 2009, as suggested on the SPLC history web page.[1] We focus on these three events because they are major events for the SPL community and already total 415 papers.[2] Therefore, we think that this sample is representative of the activity in the field, which is confirmed by [9].

From those 415 papers, a first short list of 29 papers[3] was established based on their titles. This short list was obtained after a filtering based on a disjunction of three search criteria. The first criterion aims at capturing papers that study FDs and their applications, e.g. product configuration, in real-life examples. In order to avoid papers that only consider toy examples, we included a set of terms that usually refer to real cases. They are detailed below. The second and third criteria broaden the scope by considering papers that discuss the application of SPLs in practice but which title do not contain an explicit reference to a topic different from variability modelling (e.g. architecture modelling). Note that given the nature of these latter two criteria, the search of papers was done manually. To be accepted, the title of the paper had to match at least one of the criteria, i.e. contain:

- **(Criterion 1)**
  at least one of the terms *feature diagram, feature model, variability model, configuration or derivation*
  and
  the name of a company or a reference to a practical application (e.g. *industry, practice, case study, empirical or experience*) or a reference to an economic context (e.g. *market or finance*);
- **or (Criterion 2)**
  one of the terms *product line* or *product family*
  and
  the name of a company or a reference to a practical application or a reference to an economic context
  and
  no reference to a topic different from variability modelling;
- **or (Criterion 3)**
  the name of a company or an industrial application domain
  and
  no reference to a topic different from variability modelling;

The 29 papers that matched the criteria are listed in Table I and sorted by criterion. Note that one paper ([11]) matched more than one criteria.

[1] http://splc.net/history.html

[2] Apart for SPLC'09 and the three editions of VaMoS for which we directly investigated the proceedings, the lists of published papers was retrieved from DBLP on the 20th of October 2009.

[3] Unfortunately, the access to paper [10] was not granted.

TABLE I
DISTRIBUTION OF THE SELECTED PAPERS BY CRITERIA.

Criterion	Number of papers	References
1	9	[10], **[11]**, **[12]**, **[13]**, **[14]**, **[15]**, **[16]**, **[17]**, **[18]**
2	20	**[19]**, [20], **[11]**, **[21]**, **[22]**, [23], [24], [25], [26], [27], [28], [29], [30], **[31]**, [32], **[33]**, **[34]**, **[35]**, [36], **[37]**
3	1	[38]

The set of 29 papers was then further filtered based on their abstracts and introductions. Only those whose abstract or introduction addressed the evaluation or application of variability models in industrial settings, or discussed the applicability of FD-based configuration systems were kept. At the end of the filtering process, 16 papers remained (bold-faced in Table I), i.e., roughly 4% of the total number (415) of papers.

At this point we did not know what results to expect when reviewing the papers. Therefore, data extraction was performed in an *ad hoc* way (by collecting notes of possibly relevant information), instead of using a data extraction form that we should have defined *a priori*, as suggested by Kitchenham [7]. Similarly, we defined the paper categories, which are discussed in the next section, after having read the papers.

III. FINDINGS

A. Classification of the papers

Our readings lead us to identify four categories. The first one relates to successful applications of FDs for which some adaptations to the FD language were made. The second one denotes successful applications for which no adaptations to the language were made. The third one gathers cases where the authors do not actually use FDs but acknowledge that they could have helped. The fourth one discusses unsuccessful applications. A last category was added to gather false positives, i.e., papers for which the applications of FDs turned out to be missing or too vague to tell anything about their fitness. The distribution of the papers we reviewed is summarised in Table II.

TABLE II
DISTRIBUTION OF THE REVIEWED PAPERS BY CATEGORIES.

Category	Number of papers	References
Successful applications *with* adaptations	2	[17], [16]
Successful applications *without* adaptations	4	[37], [33], [31], [14]
FDs could have helped	3	[11], [15], [13]
Unsuccessful applications	2	[22], [12]
False positives	5	[35], [34], [21], [19], [18]

B. Description of the papers

We now describe the content of all the papers belonging to each of the categories defined above.

Successful applications of FDs *with* adaptations. Gillan *et al.* [17] report on their application of an experimental FD notation applied to the modelling of embedded software in the telecommunication domain. The three main challenges to the adoption of FDs were (1) the management of the growing number of variation points, (2) the need to implement features formerly available in the software into the hardware, and (3) the specification of feature behaviours. For each of these challenges, they propose extensions to FDs suited to the telecommunication domain. The design of their system architecture is based on this extended version of FD language.

Reiser *et al.* [16] aim at specifying a unified FD language based on FODA [3]. According to their experience in the automotive industry, they claim that the biggest threats to using FDs in complex SPLs are the heterogeneity of development techniques and the divergence of subsystems. The framework they propose to address these threats uses hierarchically organised FDs linked to the other artifacts. In addition, they present a list of seven requirements for a unified language collected from their experience in the automotive industry. They also touch upon two potential benefits of FDs in this context: (1) features can link management, marketing and development within and between companies, and (2) FDs provide a central view of variability over a vast range of artifacts.

Successful applications of FDs *without* adaptations. Dordowsky *et al.* [37] discuss the adoption of SPL principles to manage the software variants and technology variations in complex avionic systems. One of their achievements is the efficient production of source code for different variants based on a FD. Unfortunately, very little detail about the FD and the way code is generated from it are available. They also report that there is a pressing need for higher integration between the FD and the other constituents of the tool chain.

Jensen [33] looks into the derivation of products in the domain of intelligence planning, collection and analysis for the US government and its allies. The author reports that FODA [3] helped to define the domain analysis process but gives no indication how FODA actually contributed to the final design. The author also complains about the lack of tool support to handle variation points across the tool chain.

Steger *et al.* [31] report on the introduction of SPLE at Bosch Gasoline Systems. They used FDs to model the variability of the SPL but do not provide feedback about the actual advantages of FDs in their application, though they mention the lack of available tool support. By applying feature analysis their goal was to reduce resource consumption, a critical aspect in embedded systems.

Wenzel *et al.* [14] explain how FDs can be used to tailor the databases of the configuration management system available in the IBM Tivoli suite. In their case, FDs proved to simplify and ease the understanding of the stakeholders who lacked specific knowledge about the underlying structure of the databases. Furthermore, using FDs for configuration reduced the *"error-prone elicitation of requirements"* and enabled the automation of choice propagation. A quick evaluation of their approach revealed that (1) the tree-based navigation coupled with cross-cutting constraints *"were regarded as a significant advantage"*, (2) FD-based database specification has a *"time-saving potential"* and (3) the reduced amount of knowledge needed to understand the database can potentially increase customer acceptance. About this latter point though, they mention that experts might miss information intentionally left out of the FD.

Applications where FDs could have helped. Deelstra *et al.* [11] studied two large industrial organisations and identified a collection of product derivation problems, including the lack of hierarchical structuring of variation points, and the lack of formal representation of variation points and dependencies. The authors clearly refer to FDs as a potential solution to the former but do not elaborate further on how to solve the other problems.

O'Leary *et al.* [15] compare two approaches for product derivation in industrial settings. A result of their comparison is a list of lessons learned for product derivation. A strong emphasis is put on the need to represent variability differently according to the type of stakeholder who is dealing with it. The authors do not rule out FDs as a potential representation but they add that FDs should not be the only one available.

Schmid *et al.* [13] studied three existing configuration tools unrelated to SPLE and compared them according to the variability concepts usually present in FDs. Although their investigation does not allow to draw any conclusion regarding the application of FDs, they provide evidence that the FD and configuration tool worlds overlap and *"co-evolved in a similar way"*.

Unsuccessful applications of FDs. Ishida [22] discusses the application of SPLE at the Nomura Research Institute to develop semi made-to-order software packages. The author rejects FD-based approaches as they *"may produce more problems than benefits"* because of the *"degree of software intensiveness* [sic]*, the ambiguity of criteria to decompose systems into features, and the frequency of requirement specification changes."*. The author adds that the key to success is abstraction rather than massive configuration of concrete artefacts. Their approach follows the model-view-controller (MVC) decomposition, and is based on a combination of UML, entity-relationship diagram and the Turbine web application framework.

Tolvanen *et al.* [12] identify approaches to define domain-specific modelling (DSM) languages that enable automated product derivation in practice. According to them, FDs are clearly not a good candidate as they *"operate at a level too general to identify DSM concepts"* to be included in the language and *"do not capture the dependencies and constraints required to define modelling constructs"*. Instead, they recommend to work at the level of the architectural model, which better supports the identification of product concepts and their relationships.

False positives. Carbon *et al.* [35] focus on the integration of SPLE principles in existing workflows and infrastructures to facilitate the customization of office devices without referring to any variability model in particular. Habli *et al.* [34] elaborate mainly on the role and definition of an appropriate configuration management plan to develop products. Although relevant to the field, they do not connect them to FDs. Helferich *et al.* [21] discuss the distinction between marketed and engineered SPLs but do not discuss the impact of this distinction on FDs. Jaring *et al.* [19] identify several variability issues taken from their experience with various industrial partners and advocate a reference framework normalising the representation of variability throughout the SPL development lifecycle. They do not explicitly point to FDs as a concrete solution. Wnuk *et al.* [18] focus on the management of variability at the requirements level. Aside from a brief reference to OVMs [2], which are compared to their "product configuration specification", they do not discuss the use of variability models.

C. General observations

We start with the observations that directly address our research question:

- **few papers on the application of FDs were found.** We first observe that less than 2% of all the papers (8 out of 415) actually discuss successful and unsuccessful applications of FDs in practice. Among these 8 papers, we observe 6 successful and 2 unsuccessful reports. This very small number of applications makes it difficult to draw any conclusion, expect that it tends to confirm our impression that there are very few experience reports. The low number of unsuccessful reports is, however, more understandable as practitioners are generally reluctant to publish unsuccessful attempts.
- **few details about the usage of FDs were found.** Out of the 6 *successful* reports, only 3 ([17], [16], [14]) provide details about how FDs were used. Unfortunately, the advantages and disadvantages described are still preliminary and the gains of using FDs are not backed up by concrete evidence. The papers that suggest FDs as a *potential* solution do not really substantiate this choice either. The first paper ([22]) reporting on *unsuccessful* uses of FDs also fails to provide substantial evidence. Its observation seems to be based on speculation rather than on facts, its justification being that *"[FDs] may produce more problems than benefits"* [22]. The second case of the *unsuccessful* applications ([12]) affirms that FDs were not suited but again does not detail the evaluation process that lead to this conclusion.
- **lack of of existing material is corroborated by other sources.** For instance, a systematic review of variability management (VM) approaches published at SPLC 2009 [9] concludes:

  > *There is only little, if any, experimental or detailed comparative analysis to show the relative advantages and disadvantages of different VM approaches. That is why it would be hard to build an evidence-based guidance for selecting a VM approach for specific development situation and context. Hence, there is a vital need of conducting comparative analysis of different approaches in order to provide the practitioners with a qualified portfolio of techniques.*

  FDs being part of VM approaches, their survey also provides the insight that comparative evaluations of FDs' advantages and disadvantages are lacking, too. Deelstra *et al.* [11] and Jaring *et al.* [19] go along the same lines saying that most of the approaches meant to support product derivation *"fail to provide substantial supportive evidence"* [11]. Gillan *et al.* [17] declare that the acceptance of FDs in telecommunication requires *"more and deeper case studies"*.

Other interesting observations we made are that there is a:

- **growing interest from practitioners.** What Table II does not show is that 8 of the 11 papers reporting on applications of FDs have been published between 2007 and 2009, i.e. more than two thirds of the experiences were reported during the last two years. Even though this trend still has to be confirmed by more comprehensive studies, it seems to show a growing interest for FDs in practice.
- **lack of tool support.** 3 papers ([31], [33], [37]) complain about the lack of tool support and the weak integration among the constituents of the tool chain. Although this information does not really help to understand how FD languages should be improved, it stresses the urge to develop dependable and easily interoperable tools. If not the key to the uptake of FDs, it would at least facilitate their evaluation.
- **lack of relationships among modelling languages.** The results of Schmid *et al.* [13] provide a first feeling that the configuration tool world and SPLE have somehow evolved in parallel but with converging goals. However, they also observed that a stronger emphasis was put on the relationship between the artifacts (e.g. components and abstract features) in configuration systems than in SPLE. As for tool support, this pinpoints a trend in SPLE research to focus on a modelling language and develop model-centred reasoning without much care for interoperability. Providing more integration among languages would be an opportunity to increase the acceptance of FDs by practitioners.

Concerning the different dialects of FDs used in the successful cases, there is unfortunately not much to say either. Gillan *et al.* [17] and Steger *et al.* [31] use their own dialect and developed specific tool support. Reiser *et al.* [16] used pure::variants and seem to use the FD language proposed by Czarnecki [4] in their examples. Jensen [33] vaguely references FODA whereas Dordowsky *et al.* [37] do not mention any particular FD dialect. Wenzel *et al.* [14] used the feature modelling plug-in for Eclipse from Antkiewicz *et al.* [39]. This multitude of dialects does not allow to draw any conclusion regarding a preference for a particular language. Furthermore, the lack of justification makes it hard to understand the rationales for their selection. The only noticeable fact is that in four of the six cases, tools were either explicitly developed ([17], [31]) or used ([16], [14]).

Before concluding, we mention two additional observations that both address the relevance of our search criteria.

- **false positives after abstract and introduction.** Table I shows that, out of the 9 papers matching the first criterion, 8 were selected for review, i.e., we had only one false positive. The second criterion is, however, far less discriminant as only 9 of the 20 (i.e. half of the papers) were selected for a complete review. This is probably due to the broader scope of this latter criterion. The third criterion turned out to be inconclusive as the only paper matching this criterion was not selected for a full review.
- **false positives after full review.** The filtering based on the abstract and introduction still resulted in 5 false positives, i.e., only 11 papers turned out to be relevant to our research question, as shown in Table II. Here, it is interesting to note that, among these 5 papers, 4 matched the second criterion whereas only 1 matched the first. This confirms the previous observation that the second criterion is too general.

The outcome of our review provides a fairly disappointing answer to the research question. It shows that for three of the most important venues in the field, the available material is sufficient neither to convince of the relevance of FDs nor to let practitioners evaluate whether FDs are a suitable solution to their problems. It also reveals that the current lack of tool support and interoperable languages might be major barriers to the acceptance of FDs in industry. Yet, the recent growing interest of practitioners should encourage researchers to actively publish their experience report to constitute a strong body of knowledge.

IV. Discussion

We now discuss the threats to the validity of our preliminary review, and the opportunity and feasibility of conducting a full systematic review.

A. Threats to validity

In Section II, we referred to our review process as *semi-systematic*. According to Kitchenham [7], a *systematic* review should follow a two-step process. The first step is the *planning* of the review. During planning, the objective and the review protocol are defined. In our case, the objective was intentionally modest as our study is meant to be a pilot, and hence more of an exploratory nature than an exhaustive collection of evidence. Consequently, our review protocol is a simplified version of the one suggested by Kitchenham, i.e. no quality assessment was performed, the data extraction protocol was not systematic and no detailed metadata analysis was carried out. Also, the protocol was not reviewed by external experts.

It is during the second step that the review is actually *conducted*. In our lightweight review process, we considered a limited set of paper sources, that were reviewed by only one researcher (the first author), without any external expert being involved in this task. In addition, all the paper venues were considered to be of equal importance during our analyses. Finally, no predefined data collection forms were used to record the results.

The impact of these limitations on the quality of our analyses is hard to tell given the small sample space we considered. However, the goal of this review was to study the feasibility of a full systematic review. The conclusions we drew previously should therefore be seen as preliminary, too.

B. Opportunity and feasibility of a full systematic review

The findings reported in the previous section point to a lack of experience reports for FDs in three important publication venues. We also observed that, in the experience reports, the justifications for the claimed advantages and drawbacks of FDs are often quite thin. Our preliminary findings might lead to the conclusion that academic research on FDs is out of touch with reality in software engineering. Several possible reasons for this lack of experience reports are conceivable. First, FDs are used but practitioners barely report on their applications. Possible causes are that practitioners do not want to publish their FDs, do not want to pay their engineers to write papers discussing their experience with FDs, or do not have sufficient incentive for publishing their experiences. Secondly, practitioners do not want to advertise unsuccessful applications of FDs. Finally, FDs might simply not be used by practitioners.

Confirming this finding is important. This justifies the need for a thorough and systematic review of a broader scope. We envision several ways to broaden the scope of our research. Obviously, we could consider more software engineering venues and journals, including those not specifically dedicated to SPLE, like ICSE, RE, ASE, TOSEM or TSE. Similarly, we could broaden the scope to include industrial venues. A completely different path would be to explore other engineering domains that also have to model variability and deal with configuration issues.

In order to increase our chances of collecting valuable results, we intend to specify a full-fledged systematic review protocol. Besides alleviating the threats to validity discussed above, we have to learn the lessons from our observations. We mentioned in our review method that the paper classification was not defined *a priori* but *a posteriori*. We now have to look at the classification we defined from a distance and evaluate how its refinement could facilitate the analysis of the results. Section III-C clearly showed that the second criterion was too general and lead to many false positive papers. Note that it was only applied to events dedicated SPLE and, thus, had a limited relevance. Further investigations are still needed to tell whether a broader scope of research will justify the refinement of this criterion. In contrast, the third criterion will probably be too general for papers outside the SPL community and will have to be refined.

The review of papers from other domains is likely to call for a completely different review protocol since its goal would be to identify well accepted techniques similar to FDs (rather than identify usages of FDs in other domains). Therefore, another pilot study will probably be required to first identify the techniques used that are comparable to FDs. Based on these results, another systematic review could identify those that are used in practice in their respective domains.

In case a systematic review confirms the preliminary results observed here, it should probably be followed by an investigation of the reasons for the absence of experience reports (e.g. conduct a large-scale survey of practice in industry).

V. Conclusion

In this paper, we questioned the availability of evidence supporting the fitness of feature diagrams in practice. In order to answer this question, we conducted a semi-systematic literature review. This review focused on two of the major venues for software product line research, i.e. the software product line conference (SPLC) and the workshop on variability modelling of software-intensive systems (VaMoS). Our preliminary findings demonstrate a lack of solid evaluation of the impact of feature diagrams on the industry. The review, however, is still preliminary, semi-systematic and limited in scope. Its results, although negative, are encouraging and call for a more thorough and systematic review, which is planned future work. In the meantime, we hope to encourage more empirical research on FDs and urge researchers to publish existing empirical results.

Acknowledgements

This work is sponsored by the Interuniversity Attraction Poles Programme of the Belgian State, Belgian Science Policy, under the MoVES project, and the FNRS.

References

[1] P. C. Clements and L. Northrop, *Software Product Lines: Practices and Patterns*, ser. SEI Series in Software Engineering. Addison-Wesley, August 2001.

[2] K. Pohl, G. Bockle, and F. van der Linden, *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer, July 2005.

[3] K. Kang, S. Cohen, J. Hess, W. Novak, and S. Peterson, "Feature-Oriented Domain Analysis (FODA) Feasibility Study," SEI, Carnegie Mellon University, Tech. Rep. CMU/SEI-90-TR-21, November 1990.

[4] K. Czarnecki, S. Helsen, and U. W. Eisenecker, "Formalizing cardinality-based feature models and their specialization." *Software Process: Improvement and Practice*, vol. 10, no. 1, pp. 7–29, 2005.

[5] P.-Y. Schobbens, P. Heymans, J.-C. Trigaux, and Y. Bontemps, "Generic semantics of feature diagrams," *Computer Networks*, p. 38, 2006.

[6] M. Mendonça, "Efficient reasoning techniques for large scale feature models," Ph.D. dissertation, University of Waterloo, 2009.

[7] B. A. Kitchenham, "Procedures for undertaking systematic reviews," Computer Science Department, Keele University (TR/SE-0401) and National ICT Australia Ltd (0400011T.1), Tech. Rep., 2004.

[8] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *Journal of Systems and Software*, vol. 80, no. 4, pp. 571 – 583, 2007.

[9] L. Chen, M. Ali Babar, and N. Ali, "Variability management in software product lines: A systematic review," in *SPLC'09*, San Francisco, CA, USA, 2009, pp. 81–90.

[10] A. Hein, M. Schlick, and R. Vinga-Martins, "Applying feature models in industrial settings," in *SPLC'00*. Norwell, MA, USA: Kluwer Academic Publishers, 2000, pp. 47–70.

[11] S. Deelstra, M. Sinnema, and J. Bosch, "Experiences in software product families: Problems and issues during product derivation," in *SPLC'04*, Boston, MA, USA, 2004, pp. 165–182.

[12] J.-P. Tolvanen and S. Kelly, "Defining domain-specific modeling languages to automate product derivation: Collected experiences," in *SPLC'05*, Rennes, France, 2005, pp. 198–209.

[13] K. Schmid and C. Kroher, "An analysis of existing software configuration systems," in *DSPL'09*, San Francisco, CA, USA, 2009, pp. 2–7.

[14] S. Wenzel, T. Berger, and T. Riechert, "How to configure a configuration management system - an approach based on feature modeling," in *MAPLE'09*, San Francisco, CA, USA, 2009, pp. 99–105.

[15] P. O'Leary, R. Rabiser, I. Richardson, and S. Thiel, "Important issues and key activities in product derivation: Experiences from two independent research projects," in *SPLC'09*, San Francisco, CA, USA, 2009, pp. 121–130.

[16] M.-O. Reiser and M. Tavakoli, R.and Weber, "Unified feature modeling as a basis for managing complex system families," in *VaMoS'07*, Limerick, Ireland, 2007, pp. 79–86.

[17] C. Gillan, P. Kilpatrick, I. Spence, R. Gawley, J. Brown, and R. Bashroush, "Challenges in the application of feature modelling in fixed line telecommunications," in *VaMoS'07*, Limerick, Ireland, 2007, pp. 141–148.

[18] K. Wnuk, B. Regnell, J. Andersson, and S. Nygren, "An industrial case study on large-scale variability management for product configuration in the mobile handset domain," in *VaMoS'09*, Seville, Spain, 2009, pp. 155–164.

[19] M. Jaring and J. Bosch, "Representing variability in software product lines: A case study," in *SPLC'02*, London, UK, 2002, pp. 15–36.

[20] M. Raatikainen, T. Soininen, I. Männistö, and A. Mattila, "A case study of two configurable software product families," in *PFE'03*, 2003, pp. 403–421.

[21] A. Helferich, K. Schmid, and G. Herzwurm, "Reconciling marketed and engineered software product lines," in *SPLC'06*, 2006, pp. 23–27.

[22] Y. Ishida, "Software product lines approach in enterprise system development," in *SPLC'07*, Kyoto, Japan, 2007, pp. 44–53.

[23] D. Sellier and G. Benguria, G.and Urchegui, "Introducing software product line engineering for metal processing lines in a small to medium enterprise," in *SPLC'07*, Kyoto, Japan, 2007, pp. 54–62.

[24] Y. Matsumoto, "A guide for management and financial controls of product lines," in *SPLC'07*, Kyoto, Japan, 2007, pp. 163–170.

[25] R. Kreuter, C. Lescher, A. Schreiber, and C. Schwanninger, "Applying a cost model for product lines: Experience report." in *MESPUL'08*, 2008, pp. 263–271.

[26] M. Khurum, T. Gorschek, and K. Pettersson, "Systematic review of solutions proposed for product line economics," in *MESPUL'08*, 2008, pp. 277–284.

[27] A. J. Nolan, "Building a comprehensive software product line cost model," in *SPLC'09*, 2009, pp. 249–256.

[28] Y. Takebe, N. Fukaya, M. Chikahisa, T. Hanawa, and O. Shirai, "Experiences with software product line engineering in product development oriented organization," in *SPLC'09*, 2009, pp. 275–283.

[29] R. Buhrdorf, D. Churchett, and C. W. Krueger, "Salion's experience with a reactive software product line approach," in *PFE'03*, 2003, pp. 317–322.

[30] J. Snyder, H. Lai, S. Reddy, and J. Wan, "Software product line support in coremetrics oa2004," in *SPLC'04*, 2004, pp. 188–191.

[31] 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 *SPLC'04*, Boston, MA, USA, 2004, pp. 34–50.

[32] R. Kolb, I. John, J. Knodel, D. Muthig, U. Haury, and G. Meier, "Experiences with product line development of embedded systems at testo ag," in *SPLC '06*, 2006, pp. 172–181.

[33] P. Jensen, "Experiences with product line development of multi-discipline analysis software at overwatch textron systems," in *SPLC'07*, Kyoto, Japan, 2007, pp. 35–43.

[34] I. Habli and T. Kelly, "Challenges of establishing a software product line for an aerospace engine monitoring system," in *SPLC'07*, Kyoto, Japan, 2007, pp. 193–202.

[35] R. Carbon, S. Adam, and T. Uchida, "Towards a product line approach for office devices - facilitating customization of office devices at ricoh co," in *SPLC'09*, San Francisco, CA, USA, 2009, pp. 151–160.

[36] W. J. Slegers, “Building automotive product lines around managed interfaces,” in *SPLC’09*, 2009, pp. 257–264.
[37] F. Dordowsky and W. Hipp, “Adopting software product line principles to manage software variants in a complex avionics system,” in *SPLC’09*, San Francisco, CA, USA, 2009, pp. 265–274.
[38] F. r. van de Linden and J. G. Wijnstra, “Platform engineering for the medical domain,” in *PFE’01*, 2001, pp. 224–237.
[39] M. Antkiewicz and K. Czarnecki, “Featureplugin: feature modeling plug-in for eclipse,” in *Proceedings of the 2004 OOPSLA workshop on eclipse technology eXchange*. New York, NY, USA: ACM, 2004, pp. 67–72.