Customer Co-Creation in Smart Production Ecosystems
Opportunities and Challenges for MDE

Deepak Dhungana
Siemens Corporate Technology, Siemens AG Austria, Vienna Austria

Keywords: Customer Co-creation, Open Configuration, Smart Production Ecosystem, Model Consolidation.

Abstract: The traditional role of customers as passive consumers is gradually changing and consumers are actively participating in co-creation of the products they buy. This shift in paradigm has implications on how products should be modeled and the tools around Model-driven Engineering (MDE) must consider new ways of dealing with open-innovation, thereby preserving the privacy and intellectual property rights of the product sellers. In this paper, we discuss how MDE can help in setting up a smart production ecosystem, enabling the interaction between product sellers and consumers and identify some challenges in this context. Some new research directions for MDE are outlined.

1 INTRODUCTION

With the increasing demand for individualized products, there is an ongoing struggle of the companies to deal with flexibility in product design and production processes. Many research initiatives in this area aim to revolutionize the future of industrial production, e.g., the Smart Manufacturing Leadership Coalition (SMLC)[1], the Industrial Internet Consortium[2], Industrie 4.0[3] etc. Product designers and factories are therefore exposing a high degree of flexibility yielding more choices for customers and enabling innovative ecosystems. We refer to such collaborative networks of product designers, factory equipment vendors, factory operators, and consumers of the products as *Smart Production Ecosystems*. We have described smart production ecosystems in detail in a recent publication (Dhungana et al., 2017b). In this paper we focus on customer co-creation in such a ecosystem.

In a smart production ecosystem, the traditional roles of customers and product sellers are redefined. With access to unprecedented information, customers are no longer passive consumers but active co-producers and engage in behaviors that strengthen their relationship with the product, company or brand (Piller et al., 2005). Customer co-creation is becoming increasingly popular among companies, and intensive communication with customers is generally seen as a determinant of the success of a new service or product (Gustafsson et al., 2012).

Customer co-creation can take many different forms. Typically online brainstorming, open expert communities, innovation groups or idea contests are adopted by companies to drive open innovation and engage external parties in solving internal problems (Piller et al., 2010b). In this paper, we discuss the idea of customer co-creation at the point of sales and its implications for model driven engineering (MDE).

Typically, customers select and configure the products they require using a configurator tool. Configurators rely on formal models of how products can be built, the kind of variability offered and the rules governing the composition of the parts (Sabin and Weigel, 1998). Such knowledge models are captured using models e.g., feature models (Kang et al., 1990) which are predefined, tested and released for sales purposes. As these models contain crucial information about the product (design, engineering and manufacturing details), they are typically not published or disclosed to public – a configurator application displays only the relevant information to the customers.

However, when allowing for customer co-creation the customers not only select the required features to configure the product based on the set of available options, but they may wish to add new features (request for model changes) or may wish to modify configuration rules in unforeseen ways. In doing so, the con-

[1]https://smartmanufacturingcoalition.org/
[2]https://industrialinternetconsortium.org/
[3]https://www.plattform-i40.de/

Dhungana, D.
Customer Co-Creation in Smart Production Ecosystems - Opportunities and Challenges for MDE.
DOI: 10.5220/0006731206250631
In *Proceedings of the 6th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2018)*, pages 625-631
ISBN: 978-989-758-283-7

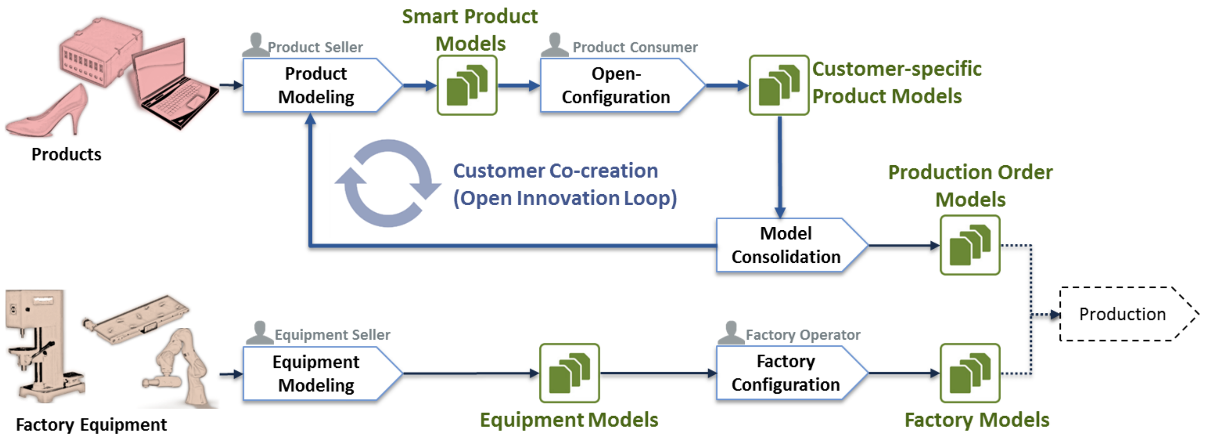

Figure 1: The role of different types of models and their interactions in a smart production ecosystem.

ventional idea of MDE to consider models as universal truth is *shaken*. The model then becomes a starting point for the customers to build upon. Models need to change at runtime, these need to be validated on the fly, and more importantly – these models are no longer hidden artifacts – as they need to be freely shared with customers. Customer co-creation therefore poses additional challenges for MDE and we discuss some of the tricky ones in this paper:

- Supporting open-innovation in a model driven smart production ecosystem.
- Supporting privacy-preserving interactions between producers and consumers.

In general, the challenges described in this paper are not specific to open-innovation or customer co-creation – these issues tend to arise in most community driven approaches, where MDE is adopted as a key enable for knowlegde representation and collaboration.

2 SMART PRODUCTION ECOSYSTEM

A smart production ecosystem is a network of factory equipment vendors, factory operators, product sellers and consumers (cf. Figure 1), interacting with each other through a common marketplace (Dhungana et al., 2017b). Interaction between stakeholders occurs through the publication and sharing of artifacts (models) to the marketplace.

In contrast to a conventional marketplace, in a smart production ecosystem, the products to be produced and the production facilities are both put on sale and the customers decide where the production should take place. Automated marketplace services are available to ensure the *producibility* of the products in selected production facilities (Dhungana et al., 2017a).

Automation and optimization of interactions in such innovative markets and networks means the different players in the ecosystem must themselves be smart, act smart and rely on the smartness of others. A smart production ecosystem relies on four pillars:

Smart Products refer to goods that are put for sale to the consumers. They are *smart* because they are not only aware of their features and materialistic properties but also *know* how they can be manufactured, thus they are aware of the requirements a factory must fulfill to produce them. Smart products are smart enough to control their own production. A *smart product model* is shared through a common marketplace.

Smart Equipment refer to devices in a factory, which publish their *production skills* as services and can be deployed as autonomous service providers in a factory. Whenever any device is put for sale, the associated equipment model describing the capabilities of the device is shared in the marketplace.

Smart Factories are configurations of smart devices which can work together, to fulfill a certain production order. Factories offer their production capabilities through a marketplace – enabling sales of *factories as services*. Smart factory models are used in combination with smart product models to check whether a certain product is producible in any given factory.

Smart Customers are consumers of the smart products. Smart customers interact with the marketplace by configuring the products they wish to buy, thereby not being limited by the set of predefined options or features. As the marketplace allows for open-innovation, the customers are partners in co-creation of the products that are offered in such a marketplace.

2.1 Sharing Knowledge through Models

Models are shared through a common marketplace, as a means to drive the creation and transfer of value through the ecosystem. Figure 2 depicts a high level meta-model used to describe/define the artifacts in the ecosystem. Several models are based on this meta-model as described below.

Equipment Models refer to formal descriptions of the factory components that can provide production capabilities in a factory. Equipment sellers publish the set of capabilities provided by their equipment as Equipment Models. Each concrete equipment is a self-contained modular unit that can execute production operations autonomously. A common ontology is shared between all the stakeholders (Capability Ontology) which is used to describe the skills of the equipment available in the marketplace (Dhungana et al., 2017b).

Factory Models refer to formal descriptions of the production facilities. A factory is seen as a specific configuration of the set of equipment deployed in its premises. In the marketplace, Factory Models are created though the selection and configuration of available Equipment Models. Typically, a factory consists of production equipment, storage systems, transport systems in a specific topology.

Smart Product Models refer to goods that are put for sale to the consumers. Products are seen as digital first-class citizens that maintain their Bill of Material (BOM), their Bill of Process (BOP), information about their variability as feature models, and a mapping between the customer view and manufacturing view depicting the mapping between feature variability and BOM variability). The materials of the product's BOM use this information to steer their own production and their step-wise transformation towards concrete product instances or product batches, i.e., smart products (Dhungana et al., 2015).

Customer-specific Product Models represent the output of the product open-configuration process carried out by the product consumers. A configurator presents the features to the consumers and captures the intent of the customer to either select/deselect an existing feature, add a new feature to the model, change existing rules, or even change the manufacturing process by changing the BOM of the product.

Production Order Models are individualized, consolidated product models that have been revised, adapted and amended after the costumers have participated in a open configuration process.

2.2 Customer Co-creation and MDE

The task of a conventional Product Configurator is to guide a customer through the derivation of a concrete product that meets their requirements from the product family representation (Sabin and Weigel, 1998). Such configurators are limited in what they offer to the customers – very often the customer may wish to have new features or variations of existing features, which may be technically feasible but not supported by the configurator at hand. This may be intentional from the perspective of product sellers, as they often try to reduce the complexity of the their portfolio and the simplify the production processes later on. But with the advancements of lot-size-one production facilities, there is often no technical reason to forbid the customer from configuring unique products - even outside the range of currently supported variants. This kind of product configuration practices are referred to as open-configuration and have been discussed in recent papers (Felfernig et al., 2014). We are therefore working on smart factories (Dhungana et al., 2017b), which can produce any product as long as the product itself is smart enough to steer its own production.

The result of a co-creation enabled configuration process is two-fold (cf. Figure 3): a formal model of the product representing partial configuration of the product required by the customer and a set of ideas/requests for additional features and attributes of the product. In a later step, the ideas/requests (submissions) are analyzed and mapped to modeling entities, so that MDE can be adopted all the way from configuration to production.

The partial product instance generated as a result of the configuration process could be sent to the factory as-is but it would reflect only a subset of the customer requirements. A in depth analysis of the co-creation submissions is needed to fulfil all the requirements.

2.2.1 Analysis of Co-Creation Submissions

As depicted in Figure 3, co-creation at the point of sales implies that the product has already been developed and it already supports most of the customer requirements. By empowering the customer to submit new requirements (e.g., new sensors, new colors, new features), there are not only benefits but also the overall task of analyzing the customer requirements and producing the unique variants becomes more challenging. Model-based requirements elicitation (Dhungana et al., 2011) can be adopted in such cases to gain contextual and enable automated analysis methods e.g., feature unweaving (Stoiber et al., 2010).

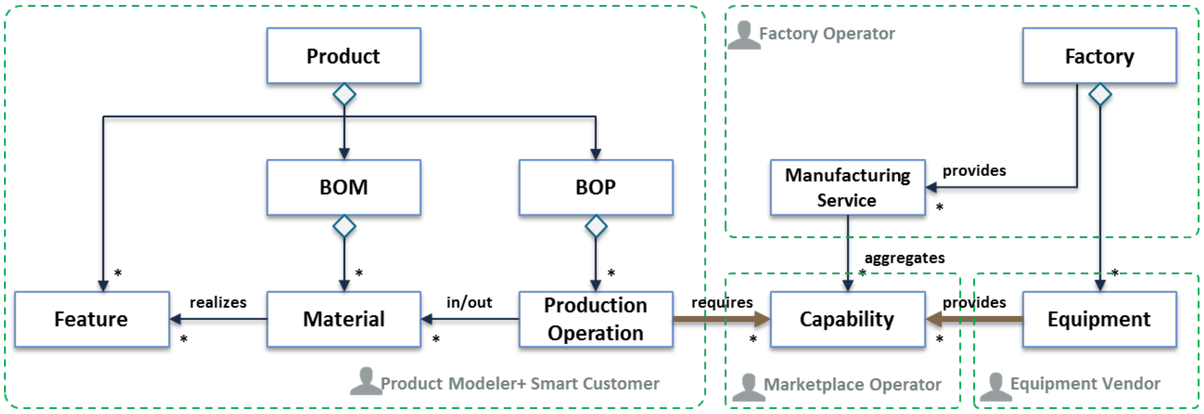

Figure 2: UML meta model of products and factories, depicting the stakeholders who create the partial models. The Marketplace Operator models the Capabilities, which are the bridge between Products, Factories and Equipment. Adapted from (Dhungana et al., 2017b).

The analysis of the requirements is to have a clear understanding of whether the new requirement articulated by the customer is feasible from a technical perspective. Given the feasibility of production and the agreement with the customer about the incurring costs, the customer-specific variant can be approved for production. In other cases, the customer-idea may be rejected.

Adoption of MDE at this phase means automated transfer of the new knowledge to the next steps in the process chain. In particular, this could be achieved by generation of partial models (comparable with model snippets (Ramos et al., 2007)) that can be merged with the formal partial model resulting from regular product configuration. Model snippets in our approach represent the differences between the offered variability of the product-line and the additional wish of the customer. Feedback (feature request from smart customers) is broken into one or more model snippets, which are then merged into one delta model (delta = difference between product model released by the owner and the merge result of the snippets resulting from a open configuration process).

2.2.2 Generation of Production Orders

A smart production marketplace enables “anytime anywhere production”. This means, the specification of how a product can be manufactured is incorporated in the product model itself. This is an integral part of the production order. In a scenario, where customers are involved in the co-creation of the product they buy, a reconciliation step is required to analyze the producibility of customer-specific variant (Dhungana et al., 2017a). Given the technical producibility of the product, production order can be generated and assigned to a factory with the required capabilities.

The challenge from the perspective of MDE is then to (semi-)automatically merge the customer-specific product model (delta model resulting from open configuration) and the partial product instance resulting from conventional configuration. Together they result in a production order - which can be autonomously handled by a smart factory. Model merging is not new to MDE and specially in product line engineering model merging has been used for managing evolution (Dhungana et al., 2008). Techniques for feature model merging (Segura et al., 2008) could also be adopted to deal with the generation of production orders.

2.2.3 Business Impact Analysis

Now that the customer-specific variant of the product has been manufactured, it is often useful to understand the market potential of this product variant. The primary goal of enabling customer co-creation is to identify features of the product that are interesting/relevant for all customers. This means, after each open-configuration meaningful variable features need to be elicited, analyzed, documented and validated. The business impact of the new features needs to be analyzed and the evolution of the reference product model can be planned. Customer involvement in co-creation does not always imply that the proposed variant of the product or the suggested new features should be automatically included in the reference product. As argued by (Bonev and Hvam, 2012), it is already a difficult task to come up with precise business calculations when using an “old-fashioned” configurator. The task of business impact analysis gets slightly more complicated with new ideas generated by customers at the point of sales.

Some support for evolution of product lines on feature level has been discussed by (Pleuss et al., 2012).

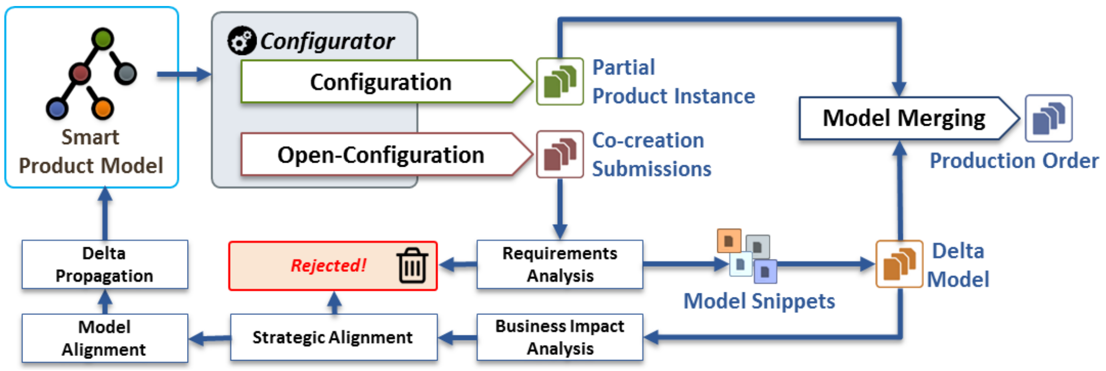

Figure 3: Modeling artefacts and processes involved in customer co-creation at the point of sales – depicting "Open Innovation Loop" supported by MDE)

Techniques such as feature unweaving (Stoiber et al., 2010) can be adopted to extract variable features from a given software requirements model. This is often referred to as product line scoping, which is the strategic decision of the business owner to include/exclude certain variants of the product in the standard portfolio (Schmid, 2002). Delta based model transformation (Diskin et al., 2011) can be helpful in propagating the change to the reference model, once the business decisions have been taken.

3 CHALLENGES AND OPPORTUNITIES

As shown in the discussion in this paper and several other papers on related topics (Dhungana et al., 2015), (Dhungana et al., 2017b), (Dhungana et al., 2017a), models form a backbone for formalizing and sharing knowledge in a smart production ecosystem. However, still a lot of work needs to be done to fully unfold the potential of MDE in this area. It is important to address these issues, as these may be some of the inhibiting factors for widespread adoption of MDE in this context, cf. embedded toolkits for user co-design (Piller et al., 2010a).

3.1 Open Innovation and MDE

One of the basic aspiration of open innovation is to harvest new product ideas/features from a larger crowd. This means open innovation relies on a *open world assumption* – whatever is not known at modeling time is not true. This is contradictory to the principles of MDE – where a closed world assumption is the basic model of operation. Extensibility of models is therefore a key requirement (sometimes meta-model extensions) to support model-based open innovation scenarios.

Innovative modeling paradigms are therefore one of the pressing needs to unfold the potential of MDE in this area. Several extensibility paradigms known to software engineering could be adopted for MDE. For example, plugin-based software extension paradigm could be adopted to define the concept of model-plugins, which could be integrated to the parent model and integrated at runtime. Feature oriented development of models (not just software) could be adopted to directly reflect the application scenarios in open-innovation. Extensibility of model interpreters is just as important as extensibility of the models for enacting the models at runtime.

3.2 Digital Rights Management in MDE

Product design specifications, factory capabilities and details of factory components typically represent crucial intellectual properties of their owners (product sellers, factory operators, factory vendors). For a smooth operation of the smart production ecosystem, all these entities must be formally modeled and shared (published in marketplaces) between the stakeholders. However, due to the crucial nature of the information, voluntary sharing of such models and information is very unlikely.

Future research in this area therefore need to consider privacy-preserving ways of modeling the information – so that the intellectual property rights are preserved but enough information is shared. Special audit methods can be adopted to ensure the production of intellectual property developed specially for one customer (Clements et al., 2013). Some other examples of items in the research agenda are:

- Encrypted sharing of models and other entities in the ecosystem and analysis operations on encrypted models (without having to fully decrypt).
- Tools and techniques for detecting violation of digital rights management based on the analysis of

published models of products, factories and factory components.

3.3 Model Synchronization vs. Knowledge Synchronization

A growing amount of companies is using model-engineering techniques (MDE) for developing and maintaining their product lines and software models. The possibilities of describing complex systems at different levels of abstraction and viewpoints seem to be especially suited for product and production models. This separation of concerns and thus, heterogeneity among the different partly overlapping models, requires an increased effort in keeping those models consistent. In complex environments, software artifacts are modeled in different languages and with different underlying meta-models. When models from different vendors of factory HW and SW components are used, the situation gets even more challenging. For the combination of or communication between those models, MDE develops model virtualization and transformation methods to support operations in a common meta-model language.

One of the most important transformation operators here is *model synchronization* (Giese and Wagner, 2009): Since several developers are working on the same product configurator model but may model it from different viewpoints (e.g., product viewpoint and factory viewpoint), changes or updates to models can happen concurrently and must be propagated to all other related models to solve potential inconsistencies. Therefore, dedicated approaches for bidirectional model synchronization are required.

A number of model synchronization approaches have been developed a in the last years (e.g., based on Triple Graph Grammars (Hildebrandt et al., 2013) or answer set programming (Cicchetti et al., 2011)), but most of these approaches are based on syntactic synchronization. Instead of focusing on model synchronization, new ways of knowledge synchronization in such ecosystems could be adopted to support multidimensional interactions in the ecosystem. There are still open research questions concerning standardization in the meta-model languages, and robustness, performance and applicability of synchronization methods in industrial environments.

4 SUMMARY AND CONCLUSIONS

This paper presented a glimpse of some of the topics of our research in the area of MDE. In our attempt to establish a full-fledged model-based solution for smart production ecosystems, we have encountered some challenges in supporting customer co-creation – which has been briefly discussed in this paper. The focus of this paper was therefore not to present a solution but rather to discuss the opportunities and challenges for MDE in the context of open innovation.

Adoption of MDE for a community based system such as the smart production ecosystem have been promising but some key features are currently difficult to implement because of the lack of mature industrial solutions to deal with privacy preserving interactions with models. Additionally, several other areas of MDE need further maturity in terms of robustness of the concepts and supporting tools, e.g., model snippet merging, synchronization of models across organizational boundaries.

REFERENCES

Bonev, M. and Hvam, L. (2012). Analyzing the accuracy of calculations when scoping product configuration projects. In *Foundations of Intelligent Systems - 20th International Symposium, ISMIS 2012, Macau, China, December 4-7, 2012. Proceedings*, pages 331–342.

Cicchetti, A., Di Ruscio, D., Eramo, R., and Pierantonio, A. (2011). Jtl: A bidirectional and change propagating transformation language. In Malloy, B., Staab, S., and van den Brand, M., editors, *Software Language Engineering*, volume 6563 of *Lecture Notes in Computer Science*, pages 183–202. Springer Berlin Heidelberg.

Clements, P. C., Krueger, C. W., Shepherd, J., and Winkler, A. (2013). A ple-based auditing method for protecting restricted content in derived products. In *17th International Software Product Line Conference, SPLC 2013, Tokyo, Japan - August 26 - 30, 2013*, pages 218–226.

Dhungana, D., Falkner, A. A., Haselböck, A., and Schreiner, H. (2015). Smart factory product lines: a configuration perspective on smart production ecosystems. In *Proceedings of the 19th International Conference on Software Product Line, SPLC 2015, Nashville, TN, USA, July 20-24, 2015*, pages 201–210.

Dhungana, D., Falkner, A. A., Haselböck, A., and Taupe, R. (2017a). Enabling integrated product and factory configuration in smart production ecosystems. In *43rd Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2017, Vienna, Austria, August 30 - Sept. 1, 2017*, pages 266–273.

Dhungana, D., Haselböck, A., and Taupe, R. (2017b). A marketplace for smart production ecosystems. In

World Mass Customization and Personalization Conference, MCPC'17, Aachen Germany, November 20 - 21, 2017.

Dhungana, D., Neumayer, T., Grünbacher, P., and Rabiser, R. (2008). Supporting the evolution of product line architectures with variability model fragments. In *Seventh Working IEEE / IFIP Conference on Software Architecture (WICSA 2008), 18-22 February 2008, Vancouver, BC, Canada*, pages 327–330.

Dhungana, D., Seyff, N., and Graf, F. (2011). Research preview: Supporting end-user requirements elicitation using product line variability models. In *Requirements Engineering: Foundation for Software Quality - 17th International Working Conference, REFSQ 2011, Essen, Germany, March 28-30, 2011. Proceedings*, pages 66–71.

Diskin, Z., Xiong, Y., Czarnecki, K., Ehrig, H., Hermann, F., and Orejas, F. (2011). *From State- to Delta-Based Bidirectional Model Transformations: The Symmetric Case*, pages 304–318. Springer Berlin Heidelberg, Berlin, Heidelberg.

Felfernig, A., Stettinger, M., Ninaus, G., Jeran, M., Reiterer, S., Falkner, A. A., Leitner, G., and Tiihonen, J. (2014). Towards open configuration. In *Proceedings of the 16th International Configuration Workshop, Novi Sad, Serbia, September 25-26, 2014.*, pages 89–94.

Giese, H. and Wagner, R. (2009). From model transformation to incremental bidirectional model synchronization. *Software and System Modeling*, 8(1):21–43.

Gustafsson, A., Kristensson, P., and Witell, L. (2012). Customer cocreation in service innovation: a matter of communication? *Journal of Service Management*, 23(3):311–327.

Hildebrandt, S., Lambers, L., Giese, H., Rieke, J., Greenyer, J., Schäfer, W., Lauder, M., Anjorin, A., and Schürr, A. (2013). A survey of triple graph grammar tools. *ECEASST*, 57.

Kang, K., Cohen, S., Hess, J., Novak, W., and Peterson, A. (1990). Feature-oriented domain analysis (foda) feasibility study. Technical Report CMU/SEI-90-TR-021, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.

Piller, F., Ihl, C., and Steiner, F. (2010a). Embedded toolkits for user co-design: A technology acceptance study of product adaptability in the usage stage. In *2010 43rd Hawaii International Conference on System Sciences*, pages 1–10.

Piller, F., Ihl, C., and Vossen, A. (2010b). A typology of customer co-creation in the innovation process.

Piller, F., Schubert, P., Koch, M., and Moeslein, K. (2005). Overcoming mass confusion: Collaborative customer codesign in online communities. 10.

Pleuss, A., Botterweck, G., Dhungana, D., Polzer, A., and Kowalewski, S. (2012). Model-driven support for product line evolution on feature level. *Journal of Systems and Software*, 85(10):2261–2274.

Ramos, R., Barais, O., and Jézéquel, J.-M. (2007). *Matching Model-Snippets*, pages 121–135. Springer Berlin Heidelberg, Berlin, Heidelberg.

Sabin, D. and Weigel, R. (1998). Product configuration frameworks - A survey. *IEEE Intelligent Systems*, 13(4):42–49.

Schmid, K. (2002). A comprehensive product line scoping approach and its validation. In *Proceedings of the 24th International Conference on Software Engineering. ICSE 2002*, pages 593–603.

Segura, S., Benavides, D., Ruiz-Cortés, A., and Trinidad, P. (2008). *Automated Merging of Feature Models Using Graph Transformations*, pages 489–505. Springer Berlin Heidelberg, Berlin, Heidelberg.

Stoiber, R., Fricker, S., Jehle, M., and Glinz, M. (2010). Feature unweaving: Refactoring software requirements specifications into software product lines. In *2010 18th IEEE International Requirements Engineering Conference*, pages 403–404.