Available online at www.sciencedirect.com

ScienceDirect

Procedia Technology 26 (2016) 447 – 454

Procedia Technology

3rd International Conference on System-integrated Intelligence: New Challenges for Product and Production Engineering, SysInt 2016

Towards Feature-based Product Line Engineering of Technical Systems

Thorsten Koch[a,*], Jörg Holtmann[a], David Schubert[b], Timo Lindemann[c]

[a]*Software Engineering Department, Fraunhofer IEM, Zukunftsmeile 1, 33102 Paderborn, Germany*
[b]*Software Engineering Group, Heinz Nixdorf Institute, Paderborn University, Zukunftsmeile 1, 33102 Paderborn, Germany*
[c]*Emmet Software Labs GmbH & Co. KG, Hoffmannstraße 12, 32105 Bad Salzuflen, Germany*

Abstract

The development of today's technical products (e.g., in automation) is characterized by high customer expectations regarding the product individualization, which causes a wide range of product variants. Original equipment manufacturers (OEMs) can apply classical approaches from product line engineering, like feature modeling, to cope with the variability and the induced development complexity. Our tool support for feature models integrates a variety of feature model extensions like feature attributes and properties, logical constraints between features and feature properties, and the distinction between features and feature realizations. Beyond that, technical products have geometrical dimensions. The OEM specifies Computer Aided Design (CAD) models to consider these geometrical dimensions and to virtually layout particular product variants. Geometrical assembly constraints specify how parts of the product can be arranged in a CAD model. However, a potential product customer cannot configure an individual product variant and virtually layout this variant in the same software tool since the respective information stems from different sources. In order to cope with this problem, we present in this paper an extension of our tool support for feature models to specify geometrical assembly constraints. Based on the proposed extension, we outline our research roadmap to consider these constraints in an online shop of an e-commerce system, in which a potential customer shall be able to configure a product variant and to virtually layout it according to the assembly constraints.

Keywords: product-line engineering, feature models, geometrical constraints

1. Introduction

Today's technical products (e.g., in automation) are characterized by high customer expectations regarding the product individualization and modularity. For example, the Pick & Place Unit (PPU) [1,2] (depicted in Figure 1) consists of four different work positions and is applicable in two different areas of application: heavy industry and pharmaceutical industry. The Stack works as workpiece input storage, and the Ramp acts as a workpiece output storage. The Stamp is responsible for labeling the workpieces, and the Crane is responsible for transporting the

[*] Corresponding author. Tel.: +49-5251-5465-127
E-mail address: thorsten.koch@iem.fraunhofer.de

doi:10.1016/j.protcy.2016.08.057

workpiece by picking and placing them between the different working positions. The PPU handles four types of cylindrical workpieces: fine-grained powder, light plastic, dark plastic, and metal. The product line of the simple automation production system considers three different setups. In the first setup, the workpieces are transported directly from the Stack to the Ramp and no additional Stamp is needed. In the other two setups, the workpieces are transported from the Stamp, and finally to the Ramp. Depending on the progressed material type, the pressure of the stamp varies.

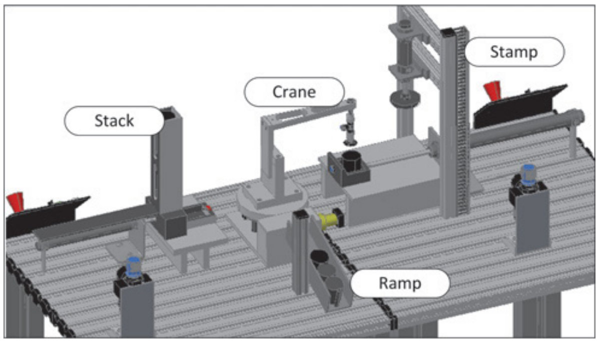


Fig. 1. Pick& Place Unit (PPU) as an example for a simple automation production system [2]

The high variability and modularity of such technical products causes a wide range of product variants. For instance, one particular PPU variant is either applicable in heavy industry or pharmaceutical industry. If the PPU variant is applied in the pharmaceutical industry, it is only allowed to handle fine-grained powder.

Original equipment manufacturers (OEMs) can apply classical approaches from product line engineering, like feature modeling [3], to cope with the variability and the induced development complexity. Our tool support for feature models integrates a variety of feature model extensions like feature attributes and properties [4], logical constraints between features [5], and the distinction between features and feature realizations. This enables the OEM to specify all product variants within a feature model. Furthermore, a potential product customer is able to configure a particular product variant and verify its correctness based on the information in the feature model [6,7].

Furthermore, technical products have geometrical dimensions. For instance, each of the particular parts Stack, Ramp, Stamp, and Crane of the PPU has a certain width, height, and depth. The OEM specifies Computer Aided Design (CAD) models to consider these geometrical dimensions and to virtually layout each particular product variant. Geometrical assembly constraints specify how parts of the product can be arranged in a CAD model. For example, the Stack cannot have the same position as the Crane and their dimensions must not overlap.

However, a potential customer cannot configure an individual product variant and virtually layout this variant in the same software tool, since the respective information stems from different sources. That is, the configuration of a concrete product variant (e.g., Stamp and Crane included in the PPU variant) influences the geometrical dimensions and correlations of its parts (e.g., the minimal distance between the Stamp and Crane must be greater than 2 meters, since otherwise the Crane would collide with the Stamp during its rotation). Furthermore, it must be ensured that the configured parts of the PPU variant do not overlap and that there is at least a small gap between them.

In order to cope with this problem, we present in this paper an extension of our tool support for feature models to specify geometrical assembly constraints. This enables the OEM to specify variability dependencies together with geometrical dependencies for his technical products in a holistic manner. Based on the proposed extension, we outline our research roadmap to consider these constraints in an online shop of an e-commerce system, in which a potential customer shall be able to configure a product variant and virtually layout it according to the assembly constraints. We illustrate the introduced concepts with the example of a product line for the PPU.

The remainder of this paper is structured as follows: In Section 2, we introduce feature modeling fundamentals. In Section 3, we introduce the current state of our tool support for the extensions of feature models for technical systems. Section 4 proposes our research roadmap towards a complete tool support. Section 5 covers related work. Finally, Section 6 concludes this paper with a summary and an outlook on open feature work.

2. Feature Modeling Foundations

Feature models have been introduced as part of the Feature-Oriented Domain Analysis to model and analyze the commonality and variability of a software product line [3]. A feature model consists of features and relationships among these features [8]. In general, a feature is an increment in the functionality of the product and provides an essential abstraction of the complex functionality under consideration.

As depicted in Figure 2, a feature model is visually represented in a tree-like structure in which nodes represent features and connections the relationship among the features. Within the feature model, connections between the features and groupings of the features represent the variability of the software product line. There are four different types of feature groups: *mandatory*, *optional*, *alternative*, and *or* [3].

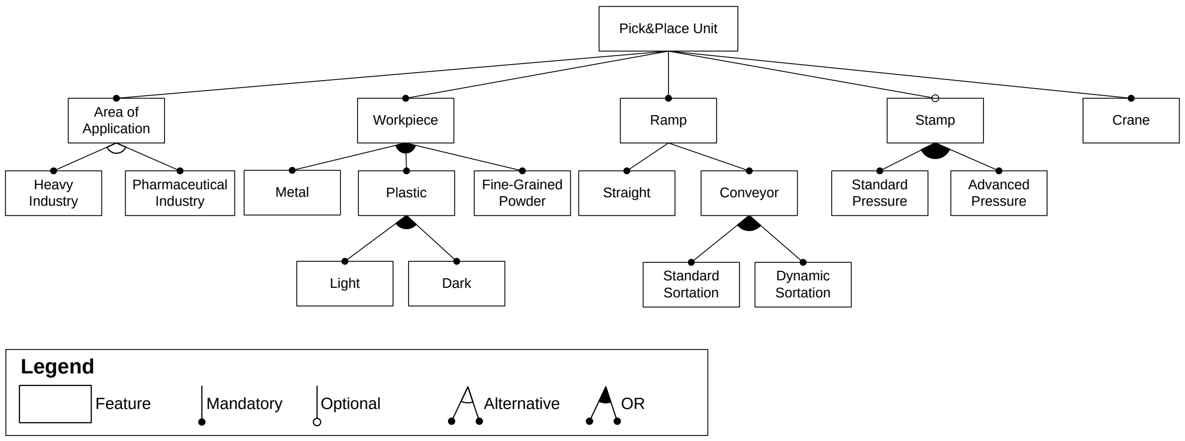


Fig. 2. Feature model for the PPU

The feature model in Figure 2 depicts an excerpt of the variability of the PPU. The left side of the feature model represents the variability in the area of application and the processed workpieces. Both features Area of Application and Workpiece are mandatory features. Thus, both features have to be selected in any valid product variant. The division of the feature Area of Application is specified by means of an alternative feature group, since the customer is only allowed to select one particular area of application. In contrast, the PPU is able to handle different workpieces in one particular product variant. Therefore, the division of the feature Workpiece is specified by means of the feature group or, meaning that the customer is able to select Fine-Grained Powder, Plastic, Metal, or all features.

Furthermore, feature models enable the specification of cross-tree constraints, such as *implies* and *excludes*, between features [5]. As mentioned in the introduction, the area of application for the PPU influences the workpieces that the PPU variant is able to handle. If the feature Pharmaceutical Industry is selected, the PPU is only able to handle fine-grained powder. We specify two cross-tree constraints (cf. Listing 1) to reflect this condition in the feature model:

Listing 1. Cross-tree constraint example for the PPU

```
Pharmaceutical Industry excludes Plastic
Pharmaceutical Industry excludes Metal
```

3. Feature Model Extensions

In this section, we describe our extensions on feature modeling for the application to technical systems. In Subsection 3.1, we present basic feature model extensions to enhance the applicability in an industrial context. In Subsection 3.2, we describe a feature model extension to enable the formalization of geometrical dependencies.

3.1. Basic Feature Model Extensions

We investigated the applicability of feature models for the specification of product lines of technical systems in an industrial context. Thereby, we learned that the base variant of feature models, as explained in Section 2, lacks expressiveness. Thus, we integrated four basic extensions to feature models, which we mainly inferred from literature. These are (i) *realizations*, (ii) *attributes and properties*, (iii) *cardinalities*, and (iv) a language to express *complex cross-tree constraints*. In the following, we refer to Figure 3, which is a more sophisticated feature model of the PPU than shown in Figure 2, to exemplify the application of these modeling extensions.

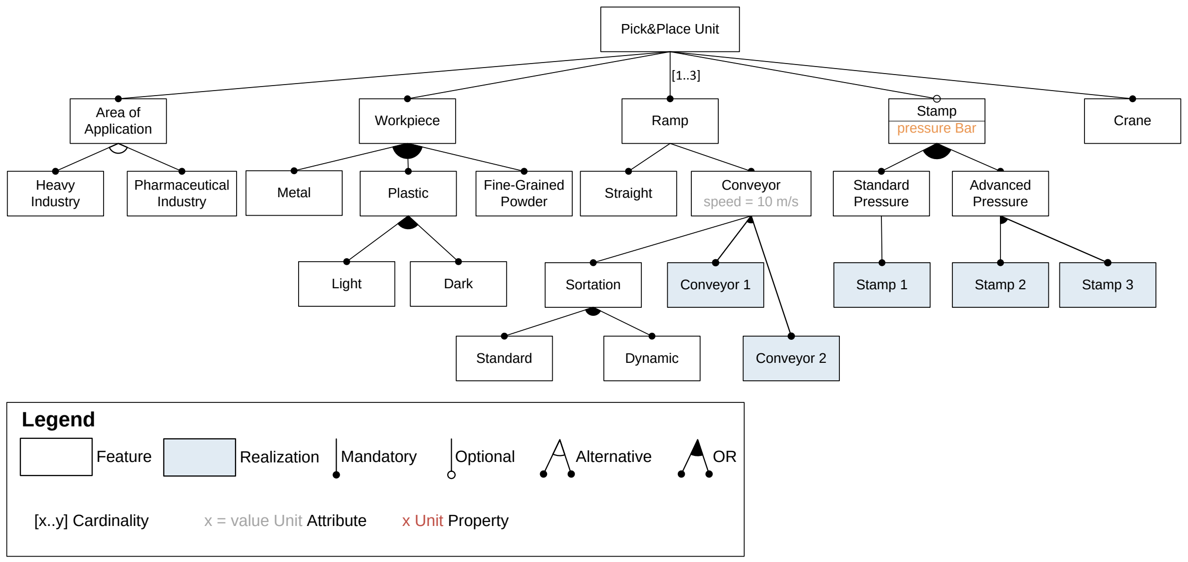


Fig. 3. Extended feature model for the PPU (cf. Figure 2)

(i) The concept *realization* enables the distinction between features of a product line and concrete technical units realizing these features. This enables to check whether concrete realizations of a feature are available (e.g., in an enterprise resource planning system). Realizations are colored in blue. Thus, Stamp 2 and Stamp 3 are realizations of the feature Advanced Pressure.

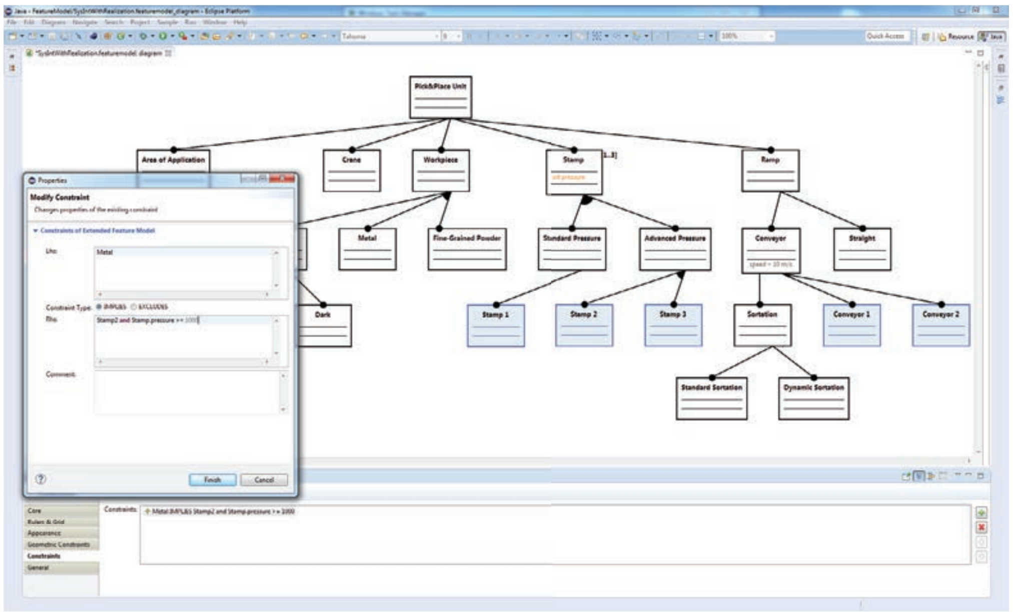
(ii) Similar to Benavides et al. [4] we use attributes to express characteristics of a feature that can be measured. As a refinement of this concept, we distinguish different kinds of characteristics and either express them in terms of properties or in terms of attributes. Properties express characteristics of a feature that are the same for all products of a product line. Thus, the value of a property is assigned within the feature model representing the product line. For example, all PPUs of our example support the same speed values (10 m/s) of their Conveyor. In contrast to this, attributes express characteristics that may differ between products of a product line. Therefore, attribute values are assigned when selecting the features for the construction of a concrete product. In our example, the pressure attribute of Stamp may have different values in the context of different products.

(iii) Cardinalities [9] express the multiplicity of features. They are depicted in square brackets ([n..m]). Here, n is the lower and m the upper bound of times the feature can be part of a concrete product. This multiplicity also encompasses all of the feature's sub-features. In our example, each PPU may have one to three slides. Thus, the Ramp feature has a multiplicity of [1..3].

(iv) We adapted the concept of using propositional logic for product line engineering [10] to express complex cross-tree constraints. Thus, we can formalize the fact that the Workpiece Metal requires a strong Stamp with a pressure of at least 1000 Bar by the expression (cf. Listing 2):

Listing 2. Complex cross-tree constraint example for the PPU

```
Metal implies Stamp 2 and Stamp.pressure >= 1000
```



Lhs: Metal

Constraint Type: [✓ IMPLIES] EXCLUDES

Rhs: Stamp2 and Stamp.pressure >= 1000

Comment:

Constraints: Metal IMPLIES Stamp2 and Stamp.pressure >= 1000

Fig. 4. Screenshot taken from our editor showing the feature model depicted in Figure 3 and the complex cross-tree constraint in Listing 2

To enable the automated analysis of feature models, feature models can be transformed into a constraint satisfaction problem. Thereby, it is possible to use off-the-shelf tools to automatically accomplish several analysis tasks like calculating the number of possible feature configurations or detecting possible configuration conflicts [6,7].

At Fraunhofer IEM, we implemented our integrated feature modeling concept by means of the plugin mechanism[1] of the Eclipse integrated development environment. The graphical editors are based on the Eclipse Modeling Framework [11] and the Graphical Modeling Framework[2]. Furthermore, our predicate-logic-based language to specify cross-tree constraints is implemented using the domain-specific language development framework Xtext [12]. Figure 4 depicts the example of Figure 3 modeled within our feature model editor.

3.2. Extension of Feature Models for the Specification of Geometrical Constraints

In this section, we describe work in progress on an extension of feature modeling to enable the specification of geometrical constraints. As mentioned in the introduction, the respective information about product variability and product geometry stems from different sources. The geometrical information stems from CAD models and is stored in an appropriate CAD system, while the logical product information is usually stored in an e-commerce application. Such an e-commerce application provides an online shop to enable the logical configuration of product variants. However, complex cross-tree constraints or the virtual layout of technical units are not part of common e-commerce applications. At Emmet Software Labs, we recently developed an extension to an e-commerce application to enable the virtual layout of product variants. To combine the advantages of both our feature modeling editor and the virtual layout capabilities of our e-commerce application, we currently integrate the different parts into one complete tool support. We reached the following two milestones in the current state of our tool support.

First, we introduced a new kind of realization, a so-called *3D-realization*, to enable the access of geometrical information stored in a 3D model within a feature model. The 3D-realization is an abstraction of a concrete technical unit and encompasses all geometrical information concerning the technical unit, like the width, the height, or the

[1] http://www.eclipse.org/pde/
[2] http://www.eclipse.org/modeling/gmp

overall boundary. This information is automatically gathered from the corresponding CAD models of the technical units.

Second, we adapted again the concept of using propositional logic for product line engineering [10], and introduce a geometrical constraint language. Inferred from literature, we distinguish three kinds of geometrical constraints [13]:

- 2D-Constraints, which are defined within a sketch
- 3D-Constraints, which are defined within a CAD file in a 3D space
- Assembly Constraints, which arrange components, assemblies, or control geometry relative to each other in product models

In the first version of our geometrical constraint language, we only covered assembly constraints, since only this kind of constraints arranges technical units relative to each other. Typical examples for assembly constraints are the allowed minimum and maximum distance between two technical units or whether two technical units intersect. By using our geometrical constraint language, we are able to formalize the constraints that the Crane and the Stamp must have a minimum distance of 2 meters; without that distance, the Crane would not be able to transport the workpieces and would collide with the Stamp:

Listing 3. Example of a geometrical constraint for the PPU

```
MinimumDistance(Crane,Stamp) >= 2m
```

This current state already enables an OEM to specify variability information together with geometrical dependencies for his technical products.

4. Research Roadmap towards the Combined Variant Configuration and Virtual Layout of Technical Systems

In this section, we introduce our research roadmap to integrate the tool support into an e-commerce system enabling a potential customer to configure a product variant and to virtually layout this variant according to the geometrical constraints in an online shop. For this purpose, we have to reach another two milestones, which we describe in the following.

First, we are going to integrate our tool support for the logical configuration of product variants and Emmet's e-commerce extension for the virtual layout of a variant. Figure 5 depicts the integration sketch. At first, a customer is able to select the technical units for this automation production system in our variant editor. If the product variant is correct, the information of the selected technical units is provided to the e-commerce extension. The CAD model of each selected technical unit is loaded from the CAD system to enable the virtual layout of the product variant. To obtain the position of each technical unit, the e-commerce extension provides this information and stores it in the variant configuration.

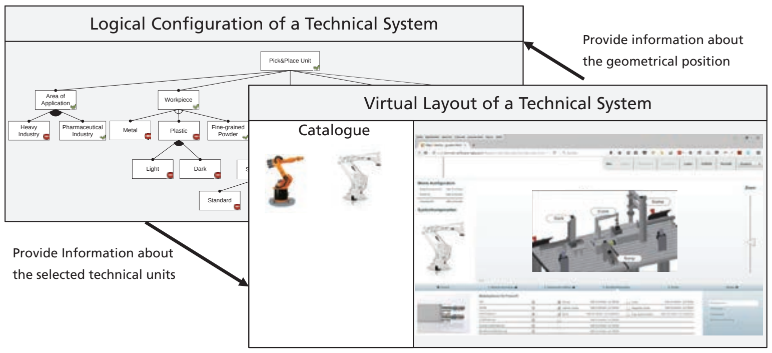

Logical Configuration of a Technical System
Provide information about the geometrical position
Pick&Place Unit
Area of Application
Workpiece
Heavy Industry
Pharmaceutical Industry
Metal
Plastic
Fine-grained Powder
Light
Dark
Standard
Virtual Layout of a Technical System
Catalogue
Provide Information about the selected technical units

Fig. 5. Sketch for the integration of the e-commerce extension and the variant configuration

Second, we are going to implement a verification of the layout against the specified geometrical constraints. Therefore, we have to implement algorithms that calculate the distance between two 3D-realizations or checks whether two lines or areas collide. The algorithm uses the layout information stored in the variant configuration and the specified constraints as input.

After reaching these two milestones, a potential customer will be able to configure a product variant and virtually layout it according to the assembly constraints.

5. Related Work

In [14], Pohl et al. propose the orthogonal variability model to represent the commonality and variability of a product line by means of variation points. A variation point represents a property of the system and is implemented by one or several engineering artifacts and is decomposed by a least one variant. Similar to feature models, variants can be optional, mandatory, or alternative and also be constrained by cross-tree constraints. However, as the basic feature model (cf. Section 2), the orthogonal variability model would lack expressiveness and the same concept must be integrated.

In [15], Brink et al. propose an approach for the specification of system families in the automotive domain. They use different variability models to specify the variability of system parts in a flexible way. The system parts stem from different engineering disciplines and might cause dependencies between the different variability models. Furthermore, they provide a tool-based algorithm for the configuration of the overall system.

Warniez and Penas et al. [16,17] present an approach that enables the specification of geometrical information in system models of mechatronic systems. They extend the Systems Modeling Language (SysML) [18] for this purpose. Like feature models, SysML is designed to capture abstract and logical information about a product but not to reflect concrete geometrical information known from CAD models. However, SysML does not provide any means to specify variability information. Furthermore, the authors do not provide tool support for any kind of analysis of this geometrical information. In contrast, we provide a research roadmap to a combined verification of the logical product variant against the feature model constraints and the virtual layout against the geometrical constraints.

6. Conclusion and Outlook

In this paper, we presented work in progress on an extension of feature modeling in our tool support. On the one hand, this includes an integration of a variety of feature model extensions needed in product line engineering of technical systems. On the other hand, this encompasses a research roadmap towards the consideration of geometrical assembly constraints in feature models as well as the current state of our tool support. The approach was illustrated with a simple automation system.

The current state of our tool support enables an OEM to specify variability dependencies together with geometrical dependencies for his technical products. Our research roadmap paves the way towards a complete tool support integrated into an e-commerce system enabling a potential customer to configure a product variant and virtually layout it according to the assembly constraints. This particularly includes the implementation of an algorithm that keeps track of the geometrical information of the product parts in the virtual layout and verifies them against the assembly constraints.

After completing the implementation of the tool support, we will evaluate our approach extensively. A potential future extension of our approach may apply a geometric modeling kernel [19], which is used within conventional CAD software tools. This approach could achieve the same algorithmic power for the verification of geometrical layouts against assembly constraints as known from CAD software.

Acknowledgment

This research is partially funded by the Federal Ministry for Economic Affairs and Energy (BMWi) under the grant ZIM and is managed by the AiF Project GmbH. Furthermore, this research is partially funded by the German Federal Ministry of Education and Research (BMBF) within the Leading-Edge Custer “Intelligent Technical Systems

OstWestfalenLippe" (it's OWL) (grant no. "02PQ1040", managed by the Project Management Agency Karlsruhe, PTKA).

References

[1] Lochau, M., Bürdek, J., Lity, S., Hagner, M., Legat, C., Goltz, U., et al. Applying model-based software product line testing approaches to the automation engineering domain. at - Automatisierungstechnik 2014;62(11):771–780. doi:10.1515/auto-2014-1099.

[2] Vogel-Heuser, B., Legat, C., Folmer, J., Feldmann, S.. Researching evolution in industrial plant automation: Scenarios and documentation of the pick and place unit. Technical Report No. TUM-AIS-TR-01-14-02; Institute of Automation and Information Systems, Technische Universität München; 2014.

[3] Kang, K.C., Cohen, S.G., Hess, J.A., Novak, W.E., Peterson, A.S.. Feature-oriented domain analysis (FODA) feasibility study. Technical Report No. CMU/SEI-90-TR-21; Carnegie-Mellon University Software Engineering Institute; 1990.

[4] Benavides, D., Trinidad, P., Ruiz-Cortés, A.. Automated reasoning on feature models. In: Advanced Information Systems Engineering; vol. 3520 of *Lecture Notes in Computer Science*. Springer. ISBN 978-3-540-26095-0; 2005, p. 491–503. doi:10.1007/11431855_34.

[5] Czarnecki, K., Eisenecker, U.. Generative Programming Methods, Tools, Applications. Boston: Addison-Wesley; 2000.

[6] Vierhauser, M., Grünbacher, P., Heider, W., Holl, G., Lettner, D.. Applying a consistency checking framework for heterogeneous models and artifacts in industrial product lines. In: 15th International Conference on Model Driven Engineering Languages and Systems (MODELS 2012). Springer. ISBN 978-3-642-33666-9; 2012, p. 531–545. doi:10.1007/978-3-642-33666-9_34.

[7] Benavides, D., Segura, S., Trinidad, P., Ruiz-Cortés, A.. Using java csp solvers in the automated analyses of feature models. In: Proceedings of the 2005 International Conference on Generative and Transformational Techniques in Software Engineering. Springer. ISBN 3-540-45778-X, 978-3-540-45778-7; 2006, p. 399–408. doi:10.1007/11877028_16.

[8] Kang, K.C., Lee, K., Lee, J.. Feature-oriented product line software engineering: Principles and guidelines. In: Domain oriented systems development – Practices and perspectives. Taylor & Francis. ISBN 0-415-30450-4; 2003, p. 29–46.

[9] Czarnecki, K., Helsen, S., Eisenecker, U.. Formalizing cardinality-based feature models and their specialization. Software Process: Improvement and Practice 2005;10(1):7–29. doi:10.1002/spip.213.

[10] Mannion, M.. Using first-order logic for product line model validation. In: Proceedings of the 2nd International Conference on Software Product Lines (SPLC 2). Springer. ISBN 978-3-540-43985-1 (Print), 978-3-540-45652-0 (Online); 2002, p. 176–187. doi:10.1007/3-540-45652-X_11.

[11] Steinberg, D., Budinsky, F., Paternostro, M., Merks, E.. EMF: Eclipse Modeling Framework. The Eclipse Series; 2nd ed.; Addison-Wesley; 2008. ISBN 978-0321331885.

[12] Eysholdt, M., Behrens, H.. Xtext: Implement your language faster than the quick and dirty way. In: Proceedings of the ACM International Conference on Object Oriented Programming Systems Languages and Applications (OOPSLA'10). ACM. ISBN 978-1-4503-0240-1; 2010, p. 307–309. doi:10.1145/1869542.1869625.

[13] Anderl, R., Mendgen, R.. Modelling with constraints: theoretical foundation and application. Computer-Aided Design 1996;28(3):155 – 168. doi:10.1016/0010-4485(95)00023-2.

[14] Pohl, K., Böckle, G., van Der Linden, F.J.. Software product line engineering: foundations, principles and techniques. Springer Science & Business Media; 2005.

[15] Brink, C., Heisig, P., Sachweh, S.. Using cross-dependencies during configuration of system families. In: 1st International Workshop on Process, Methods and Tools for Engineering Embedded Systems. Springer. ISBN 978-3-319-26843-9 (Print), 978-3-319-26844-6 (Online); 2015, p. 439–452. doi:10.1007/978-3-319-26844-6_32.

[16] Warniez, A., Penas, O., Plateaux, R., Soriano, T.. SysML geometrical profile for integration of mechatronic systems. In: Proceedings of the 2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM). 2014, p. 709–714. doi:10.1109/AIM.2014.6878162.

[17] Barbedienne, R., Penas, O., Choley, J.Y., Rivière, A., Warniez, A., Della Monica, F.. Introduction of geometrical contraints modeling in SysML for mechatronic design. In: 2014 10th France-Japan / 8th Europe-Asia Congress on Mecatronics (MECATRONICS). IEEE; 2014, p. 145–150. doi:10.1109/MECATRONICS.2014.7018580.

[18] Object Management Group (OMG), . OMG Systems Modeling Language (OMG SysML): Version 1.4, OMG Document Number: formal/2015-06-03; 2015. URL: `http://www.omg.org/spec/SysML/1.4/`.

[19] Shah, J.J., Mantyla, M.. Parametric and Feature Based CAD/Cam: Concepts, Techniques, and Applications. 1st ed.; New York, NY, USA: John Wiley & Sons, Inc.; 1995. ISBN 0471002143.