Transdisciplinary Engineering Methods for Social Innovation of Industry 4.0
M. Peruzzini et al. (Eds.)

doi:10.3233/978-1-61499-898-3-665

Glencoe – A Tool for Specification, Visualization and Formal Analysis of Product Lines

Anna SCHMITT[1], Christian BETTINGER[1] and Georg ROCK [1]
Trier University of Applied Sciences, Germany

Abstract. Manufacturers of all business areas are faced with growing customer demands and the corresponding complexity of their products. The need to offer highly customized variants becomes obvious for example in the automotive industry: innovative cars with configurable features move customers to buy newer models in increasingly shorter intervals. However, the costs are expected to be nearly stable from the customer's perspective. An industry approved approach to systematical design customizable products at reasonable costs are multivariant product lines: product line engineers compose models describing common and distinctive product features as well as logical constraints. Subsequently these models can be analyzed and optimized. In this paper, we present a free variant management tool called Glencoe. Glencoe is a web application developed and hosted at Trier University of Applied Sciences. Therefore, it is not necessary to install any software; it is accessible from any computer or mobile device. Glencoe allows the user to specify and visualize feature models using different views which emphasize specific aspects. The application is connected to several well-known and freely available theorem provers and SAT solvers. Thus, the user is able to analyze the product line for consistency, the number of product configurations and the existence of dead features. A set of metrics helps to evaluate the product line healthiness. Glencoe is intuitive to use since it is built according to state-of-the-art user interaction paradigms originating from mobile application development. As a rapid prototyping tool, it allows beginners as well as experts to build and analyze product lines.

Keywords. Variability Management, Feature Model, Product Line, Visualization, Formal Analysis

Introduction

Mass customization is the current paradigm to satisfy the growing customer demands in an economic and efficient way. It combines the benefits of industrial mass production and individual customization. Manufacturers are enabled to improve their competitiveness by keeping the quality high and the costs at a reasonable level [1] [2] [3].

Product line management is an industry approved approach to reach this goal. A product line groups all variants of a product by describing common features, which are part of every product configuration, as well as individual features, which constitute variability. Feature models are a commonplace modelling technique. They allow the

1 Corresponding Authors., Mail: {anna.schmitt, c.bettinger, g.rock}@hochschule-trier.de.

specification of joint and variant features as well as overall constraints within a product line [4] [5] [6] [7] [8].

The following points summarize the major challenges in this field:

1. comprehensible and assistive visualization of small and complex feature models (from 10 up to more than 10^8 features per model)
2. automated formal analysis of product lines
3. automated formal deduction of buildable product configurations
4. optimization of product configurations regarding one or more optimization criteria
5. definition of metrics which allow reasoning the *healthiness*[2] of a product line

Since the underlying problems are computationally hard, it is necessary to transform feature models into logical representations, apply automatic theorem provers or SAT solvers and interpret the output.

In this paper we present the variant management tool Glencoe. It supports even the inexperienced user to meet the challenges described above by making the described process transparent.

1. Glencoe

Glencoe is a free to use and responsive web application for visualization, specification and formal analysis of product lines hosted at Trier University of Applied Sciences. It is a research project started in 2012 and still under active development by a team of researchers and students. One of its advantages is the possibility to evaluate new ideas and concepts based on current research in short intervals. Therefore, Glencoe is qualified as a part of feasibility and acceptance studies on product line management in a professional context.

The following sections describe some core features of Glencoe in more detail. Standard features such as zooming, importing and exporting of feature models etc. are omitted but nevertheless available in Glencoe.

1.1. Visualization of Product Lines

Visualizing product lines with their components and complex constraints plays a vital role in systematic management of variability. Therefore, Glencoe provides different views, each of which is optimized for a distinct use case.

The *Tree* view, shown in Figure 1, applies a representation base on the FODA notation. Kang et al. introduced this notation in the year 1990 [1]. This traditional view focuses on the hierarchical relations between features. Additionally, it allows to visualize cross-tree constraints such as *required* features and *mutual exclusion* between features all at once in an intuitive manner.

[2] The notion *healthiness* integrates several product line metrics as for example the impact of one feature on the complete product line. In this way possible defects within the product line specification are already indicated during the specification phase avoiding expensive corrections in later phases.

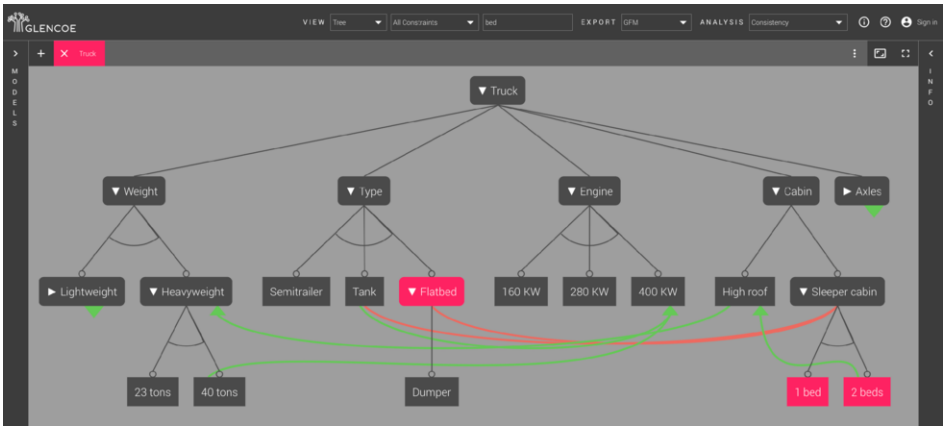

Figure 1. Tree View.

Figure 2 shows the *Icicle* view. Like the Tree view this representation maintains the hierarchical structure of a feature model. By eliminating the connecting edges and increasing the reserved space for each feature this view is predestinated for mobile devices with touch-based user interfaces.

The *Circle* view, shown in Figure 3, neglects the hierarchical nature of feature models and focuses on cross-tree constraints instead. This is achieved by positioning the features at the circumference of a circle, within which all constraints are displayed at once. This assists the user in identifying strongly connected feature sets.

The *Table* view, which is shown in Figure 4, focuses on feature properties and resembles conservative spreadsheets commonly found in companies not using dedicated product line management software.

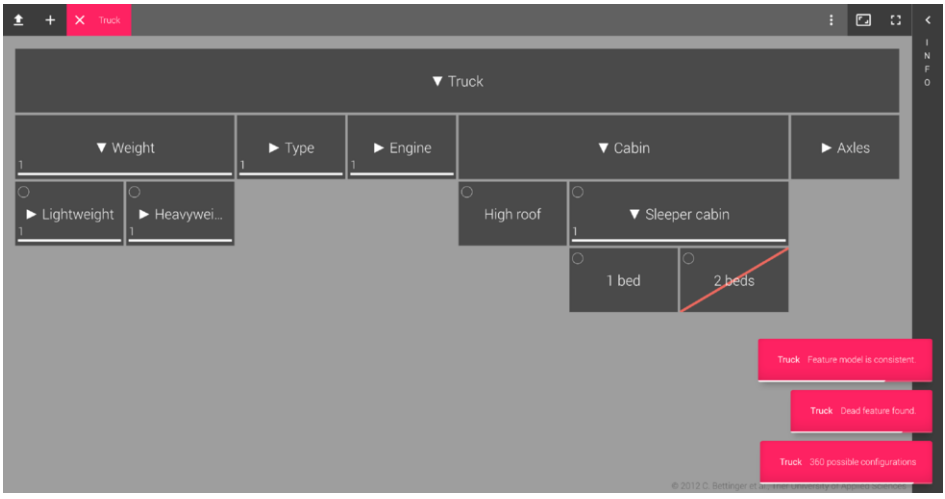

Figure 2. Icicle View.

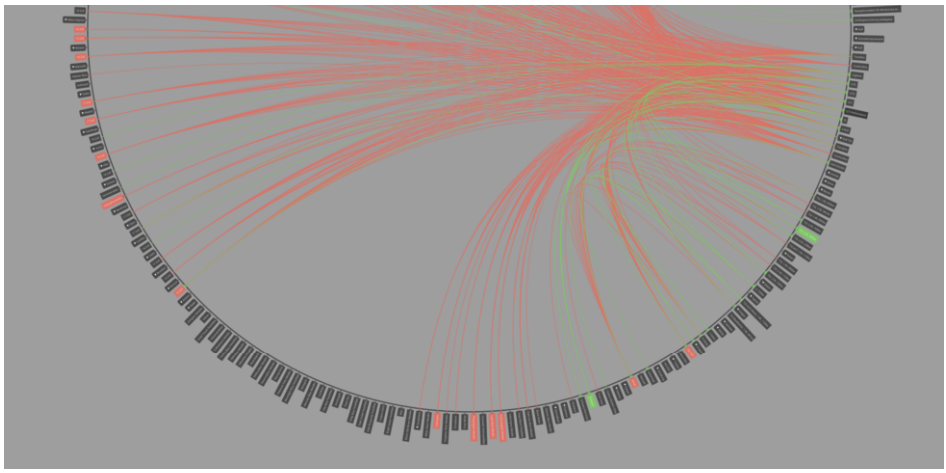

Figure 3. Circle View.

Figure 4. Table View.

In order to control the information density, the user can highlight searched features, fold certain subtrees or choose among different modes of displaying cross-tree constraints as there are:

- *All Constraints*. All constraints are displayed permanently.
- *Selective Constraints*. Only the constraints of a selected feature are displayed and the involved features are highlighted.
- *Transitive Closure*. All features that are directly or indirectly linked to a selected feature by some constraints are highlighted.
- *No Constraints*. No constraints are displayed at any time.

1.2. Specification of Product Lines

Glencoe allows creating new feature models as well as importing existing product lines from various academic or commercial file formats.

As shown in Figure 5 the user can edit feature models (this can be done in every view as described in section 1.1) by invoking the context menu on a feature. The following list gives a short overview about the possible actions:

- edit feature properties (name, optionality, type: feature, alternative, selection, selection with lower and upper bounds)
- remove a feature including all descendant features and involved constraints
- add a child feature
- add/remove a constraint (requirements, mutual exclusions)

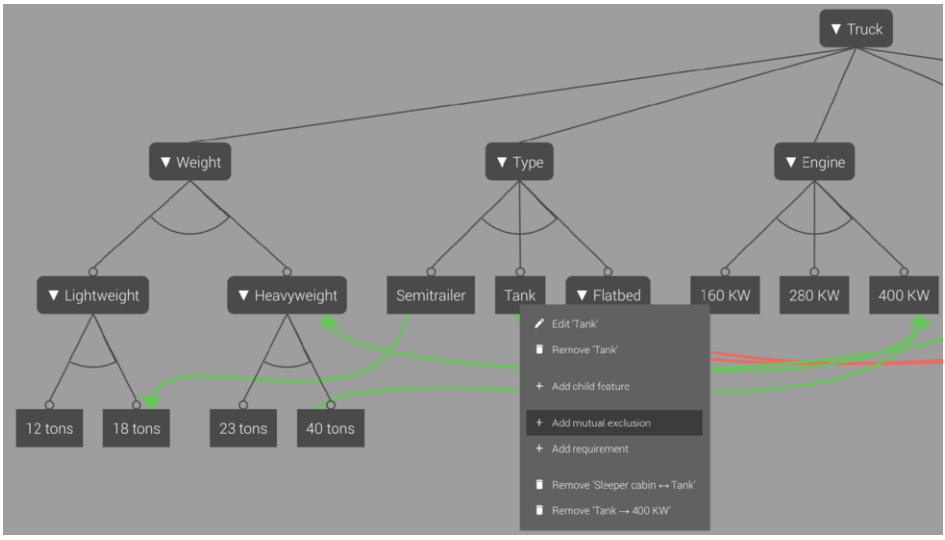

Figure 5. Editing a feature model.

1.3. Formal Analysis of Product Lines

Real-world product lines can easily contain a vast number of features and variants. Because of this magnitude and the inherent complexity of constraints and their implications on variability a manual analysis is usually not possible.

Glencoe provides an automated formal analysis of product lines. At present, it supports three methods of analyses, which determine

- the consistency of a feature model, i.e. the existence of at least one buildable product configuration,
- all dead features, i.e. features that are not part of any product configuration,
- the total number of buildable product configurations.

The underlying process, shown in Figure 6, consists of two steps: First, the feature model is transformed into a normalized propositional logic formula (CNF) utilizing the processing capacity of the client computer. The formula is transmitted to the server on each analysis request. Second, a set of freely available and well-known solvers as well as specifically designed and in-house developed solvers is available to execute the analysis. This concept of combining a browser-based web application with a server-side solver farm is a core aspect of Glencoe.

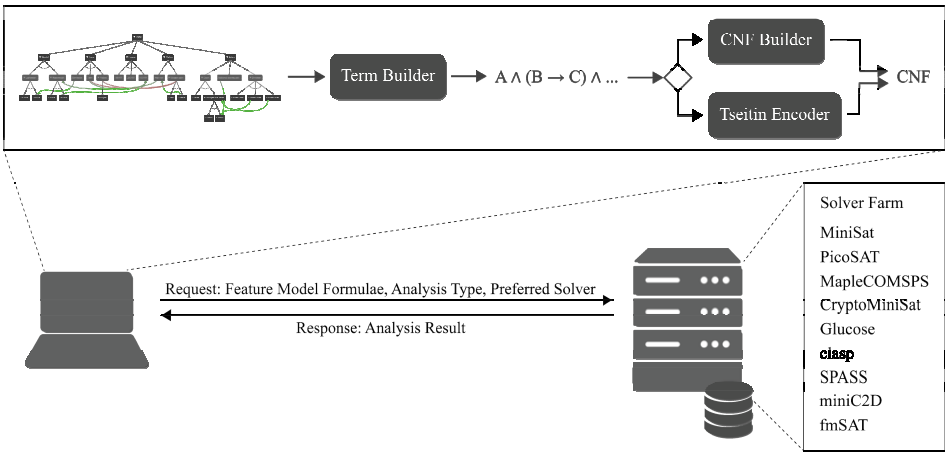

Figure 6. Call structure of Glencoe, in case of an analysis request.

While the user can choose his preferred solver, Glencoe may select a more appropriate one in terms of efficiency depending on the requested analysis. The following list includes the currently integrated solvers:

- MiniSat [9] was developed by Niklas Eén and Niklas Sörensson. The goal was to create a compact, but complete and efficient SAT solver. The implementation is based on the Davis-Putnam-Logemann-Loveland algorithm (DPLL), which was expanded by several optimization mechanisms, e.g. the use of watch lists during the unit propagation, non-chronological backtracking and conflict-driven clause learning (CDCL). MiniSat is an open-source project and served as a basis for most of the following SAT solvers.
- PicoSAT [10] is a SAT solver developed by Armin Biere at Johannes Kepler University Linz. The CDCL algorithm was further optimized by implementing a particular treatment of binary clauses, the maintenance of an occurrence list and the use of clause learning restarts.
- MapleCOMSPS [11] is a member of the MapleSAT solver series developed at University of Waterloo, Canada. The key innovation of this solver is the use of a learning rate branching heuristic (LRB). MapleCOMSPS won 1st in the main track of SAT Competition 2016.
- CryptoMiniSat [12] is developed by Mate Soos. The CNF-based input syntax DIMACS, which is expected by all common SAT solvers, is extended by XOR clauses. These clauses have a particular importance for cryptographic applications as well as for the analysis of feature models.
- Glucose [13] was developed by Gilles Audemard and Laurent Simon. Since version 4 it enables a parallel execution on multi-core CPUs with a shared memory.
- clasp [14] is an answer set solver developed at University of Potsdam. It can be used for answer set programming as well as for SAT solving.
- SPASS [15] is an automated theorem prover for first-order logic. It was developed at Max Planck Institute for Informatics. Due to its sophisticated Thoralf Skolem-awarded internal CNF transformation [16] Glencoe supports selection features with arbitrary lower and upper bounds.

- miniC2D [17] was developed by Umut Oztok and Adnan Darwiche at University of California, Los Angeles. Its algorithm uses common techniques of SAT solving for compiling CNF formulae to decision sentential decision diagrams (SDD). This enables Glencoe to determine the total number of buildable product configurations even for large product lines within very short times.
- fmSAT [18] is an in-house project of Trier University of Applied Sciences and is developed by Sandra Spang and Anna Schmitt. This CDCL-based implementation is optimized for the analysis of feature models and deduction of buildable product configurations.

1.4. Metrics

To enable the quantitative valuation of a product line Glencoe provides different metrics. They range from basic ones (e.g. number of features and constraints, model depth) to more advanced ones like *Impact* and *Transitive Closure Cardinality*. Those metrics support product line engineers in identifying strongly coupled features which may need a design review. We established the notion *healthiness* to indicate situations where there is no logical error such as a dead feature or an inconsistent product line specification, but a hint to a possible defect within the product line specification.

1.5. Cloud Integration

Glencoe allows the user to upload imported or created feature models to a remote storage service hosted at Trier University of Applied Sciences. Modifications, analysis results as well as application settings are persisted immediately without explicit user interaction.

Authentication is handled by adopting a single sign-on approach using third-party service providers, e.g. Google, Amazon or LinkedIn. Neither user credentials nor feature model data are exchanged among those two services.

2. Related Work

There are some product line management tools, like Gears [4], pure::variants [5] and FAMA [19]. The study of Pereira et al. [6] showed, most tools are realized as a plugin to more complex development environment. Glencoe follows a different approach, as it is a freely available, easy to use and easy to adapt web application.

S.P.L.O.T. [20] [21] is another browser-based product line manage-ment tool. It supports creation and editing of feature models, automated analysis and product configuration. Feature models can be exported or saved to a public repository. This tool does neither support advanced graphical visualization nor the creation of private user accounts.

Invar [22] is a web application especially for configuration of product lines. It emphasizes a unified perspective on product lines created with different modeling methods and tools. Advanced graphical visualization, editing and additional analyses are not supported.

3. Conclusion and Future Work

Glencoe is an easy to use web application for the specification, visualization and automatic formal analysis of product lines. It is backed by industry approved concepts as well as state-of-the-art approaches based on current research.

The different views assist the user to comprehend information even within complex product lines and enable the visualization of product lines in a flexible way. The integrated metrics and automatic analysis procedures provide additional support within this process.

Glencoe is under active development and can be accessed via *glencoe.hochschule-trier.de*. On the first start a demo feature model is loaded automatically in order to get new users started immediately.

Glencoe facilitates professional companies to experiment with profound variability management methods with a minimal effort and risk. As Glencoe is based on well-established user interaction principles, the gap between conservative spreadsheet-based approaches and formal management techniques is easy to overcome.

Nevertheless, a lot of challenges remain for the future work: The integration of an interactive product configuration module provided by the in-house developed fmSAT solver is the next major step. This module allows the user to manually deduct valid products by selecting or deselecting product features in an arbitrary order. During the configuration process fmSAT works in background and takes care that all the constraints are met.

Another step towards variability of architecture descriptions is the realization of a variability dependent configurable function net including its visualization using a special view containing function blocks, corresponding ports and communication lines.

At present, Glencoe supports various common import and export file formats for feature models. In order to improve interoperability with commercial applications additional sets of importers/exporters are planned especially to support the description of variability within requirements specifications.

All the mentioned components are currently work in progress and expected to be released this year.

References

[1] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak and A. S. Peterson, *Feature-Oriented Domain Analysis (FODA) Feasibility Study*, Carnegie-Mellon University, 1990.

[2] K. Pohl, G. Böckle, F. J. Van der Linden, *Software Product Line Engineering: Foundations, Principles and Techniques*, Springer-Verlag New York, 2005.

[3] S. Klawitter, G. Rock, Handling of Product Variability Modeling Approaches, In: J. Stjepandić et al. (eds.) *19th ISPE International Conference on Concurrent Engineering (ISPE CE)*, Trier, Germany, 2012, Springer-Verlag, London, 2013, pp. 169–179.

[4] C. W. Krueger, Multistage Configuration Trees for Managing Product Family Trees, In: *Software Product Line Conference (SPLC)*, Tokyo, Japan, 2013, pp. 188–197.

[5] Pure-systems GmbH, *Variant Management with pure::variants*, White Paper, Magdeburg, 2006.

[6] J. A. Pereira, K. Costantino, E. Figueiredo, A Systematic Literature Review of Software Product Line Management Tools, In: *Software Reuse for Dynamic Systems in the Cloud and Beyond*, Springer International Publishing, Miami, FL, USA, 2014, pp. 73–89.

[7] T. Thüm, C. Kästner, F. Benduhn, J. Meinicke, G. Saake, T. Leich, FeatureIDE: An Extensible Framework for Feature-oriented Software Development, *Science of Computer Programming*, Vol. 79, 2014, pp. 70–85.

[8] G. Rock and S. Mann, Dealing with Variability in Architecture Descriptions to Support Automotive Product Lines, In: *Workshop on Variability Modelling of Software-Intensice Systems (VAMOS)*, Sevilla, Spain, 2009, pp. 111–120.
[9] N. Eén, N. Sörensson, An Extensible SAT-solver, In Giunchiglia E., Tacchella A. (eds.): *Theory and Applications of Satisfiability Testing*, Springer, Berlin, 2004, pp. 502–518.
[10] A. Biere, PicoSAT essentials, *Journal on Satisfiability, Boolean Modeling and Computation (JSAT)*, Delft, Netherlands, 2008, pp. 75–97.
[11] J. H. Liang, C. Oh, V. Ganesh, K. Czarnecki and P. Poupart, MapleCOMSPS, MapleCOMSPS_LRB, MapleCOMSPS_CHB, In: *International Conference on Theory and Applications of Satisfiability Testing (SAT)*, Bordeaux, France, 2016, pp. 52–53.
[12] M. Soos, Enhanced Gaussian Elimination in DPLL-based SAT Solvers, In: *Federated Logic Conference (FLoC)*, Edinburgh, Scotland, UK, 2010, pp. 2–14.
[13] G. Audemard, L. Simon, Lazy Clause Exchange Policy for parallel SAT solvers, In: *International Conference on Theory and Applications of Satisfiability Testing (SAT)*, Vienna, 2014, pp. 197–206.
[14] M. Gebser, B. Kaufmann, T. Schaub, Conflict-Driven Answer Set Solving: From Theory to Practice, In: *International Conference on Artificial Intelligence (IJCAI)*, Beijing, China, 2013, pp. 912–918.
[15] C. Weidenbach, D. Dimova, A. Fietzke, R. Kumar, M. Suda, P. Wischnewski, SPASS Version 3.5, In: *International Joint Conference on Automated Deduction*, Montreal, Canada, 2009, pp. 140–146.
[16] A. Nonnengart, G. Rock, C. Weidenbach, On Generating Small Clause Normal Forms, In: *International Conference on Automated Deduction (CADE)*, Lindau, Germany, 1998, pp. 397–411.
[17] U. Oztok, A. Darwiche, A Top-Down Compiler for Sentential Decision Diagrams, In: *International Joint Conference on Artificial Intelligence (IJCAI)*, Buenos Aires, Argentina, 2015, pp. 3141–3148.
[18] S. Spang, *Development and Evaluation of Enhanced Procedures for Formal Analysis of Feature Models*, Thesis, Trier, 2017.
[19] D. Benavides, S. Segura, P. Trinidad, A. Ruiz-Cortés, FAMA: Tooling a Framework for the Automated Analysis of Feature Models, In: *First International Workshop on Variability Modelling of Software-Intensive Systems (VAMOS)*, Limerick, Ireland, 2007, pp. 129–134.
[20] M. Mendonca, M. Branco, D. Cowan, S.P.L.O.T. - Software Product Lines Online Tools, In: *OOPSLA*, Orlando, Florida, USA, 2009, pp. 761–762.
[21] M. Mendonca, M. Branco, D. Cowan, 2009, *S.P.L.O.T.*, Accessed: Jan, 21 2017, http://splot-research.org
[22] D. Dhungana, D. Seichter, R. Rabiser, P. Grünbacher, D. Benavides and J. A. Galindo, Configuration of Multi Product Lines by Bridging Heterogeneous Variability Modeling Approaches, In: *Software Product Line Conference (SPLC)*, Munich, Germany, 2011, pp. 120–129.