IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-6, Issue-4 E-ISSN: 2347-2693

Software Product Line Configurations Generation using Different Types of Tools – A Comparison

A. Saini[1*], Rajkumar[2], S. Kumar[3]

[1]Department of Computer Science, Gurukula Kangri Vishwavidyalya, Haridwar, Uttarakhand, India
[2]Department of Computer Science, Gurukula Kangri Vishwavidyalya, Haridwar, Uttarakhand, India
[3]Department of Computer Science, Gurukula Kangri Vishwavidyalya, Haridwar, Uttarakhand, India

[*]*Corresponding Author: ashish_saini66@yahoo.com, Tel.: +91-8791158699*

Available online at: www.ijcseonline.org

Received: 10/Mar/2018, Revised: 17/Mar/2018, Accepted: 29/Mar/2018, Published: 30/Apr/2018

Abstract— Feature model's analysis is a booming area and should be automated is a thriving research topic, an area of attraction for both practitioners and researchers from last two decades. Meanwhile, a number of methods and tools facilitate to increase the analysis of feature models and also check complexity of feature model. As numerous of tools are given by researchers and practitioners, but why a tool is used and for which purpose, it's a basic problem and creates a blurry scenario and this blurriness generate hurdles to select an analyzing tool to analyze a feature model. To clear this picture, we present a paper, where we compare four analysis tools (FeatureIDE, SPLOT, FaMa and BeTTy) on the basis of some fundamental factors (Availability of Tool, Cross Tree Constraint, Support Testing, Fault Detection, Product Generation, Statistics of Model and Model Composer). The comparison will show in form of a table at the end of this paper, through which users can choose a tool for their work.

Keywords— Software product line, Feature models, Automated analysis, Testing, Validation

I. INTRODUCTION

A Product line or software product line a new and intensive area in software engineering. SPL is a line in which product share set of common features to gratify the exact needs of the specific market section. The focusing part of the SPL is to developing a set of communal assets which contain design, test plans, test cases, change requirements, reusable software components, better management and other artifacts [1][13]. Big companies like Dell, Samsung, Motorola, Hewlett-Packard and other complete their product on the basis of SPL concept.

The main concept of Software Product Line is its feature model. This model used by companies to represent the common and variable feature in Software Product Line. A feature of any product demonstrates the functionality increment relevant to some stakeholders. A big advantage of Software Product Line is to achieve increment in reusability of a feature in numerous products of SPL.

Practically, a product in SPL developed by feature modeling that represents different lookouts and another subsystem. According to company's point of view, to achieve a good product with SPL it is compulsory to have tools which show the variability management, test cases, test stages and other required field, tools represent the same product in the number of ways [2].

Moreover, tools support for SPL should support the complete development process, not for some or particular activities. It's a complex task to choose a tool that best for SPL development because there are numerous tools introduce in previous years and available as open source in the market and they all have their own specifications and functionality [3]. In the same context, this paper introduces some SPL development tool and compare them on their working as well as production design on different parameters.

Further, this paper design as in the second section presents a brief introduction to all four tools (FeatureIDE, SPLOT, FAMA, and BeTTy). Section three based on analysis of a feature model design with all four tools and discuss their features. In Fourth section compare all the tools on the basis of result which find in previous (third) section. Finally, the fifth section concludes the paper with future perspectives and importance of tools in Software Product Line Engineering.

II. TOOLS FOR FEATURE MODEL

A. FeatureIDE

In product line engineering it's available as an open source framework and to create any product it starts with feature modeling, domain analysis and also covers other areas like designing, implementation and maintenance with supporting compatible feature house/articulate tools like FOSD(Feature-

Oriented Software Development), AHEAD(as default) model composer suit. On the other side, featureIDE doesn't support only one language, it supports different compatible tools with different languages like Java, C, C++, C#, JavaCC, Haskell and XML [6].

B. S.P.L.O.T

It is a web-based open source tool which developed by Marcilio Mendonca in 2008. Generally, this tool uses to analysis feature models including validity valid configurations detection of lend features and many more, which help not only industries also in the research field. The time-consuming task in SPL is variants (selecting/deselecting of features) in a feature model. In this context, SPLOT provides a semi-automatic configuration tool that reproduces decision (selecting/deselecting) automatically to abide constraints in the model [10].

C. FAMA

FeAture Model Analyzer (FAMA) is a framework which automated analyze a feature model on the basis of reasoners. FAMA amalgamate reasoners (default reasoners, Choco, Sat4j, JaCop, JavaBDD) designed by the third party, that boost the functionality of the FAMA. For best results, it automatically selects the adequate solver at runtime as the operation considers by the user. It can accomplish the task of both types of feature models basic as well as extended. Numerous versions of FAMA work for design and analyze released for SPL [5] [7].

D. BeTTy

BeTTy (BEnchmarking and TesTing on AnalYsis) is another tool to automatically analyze and detect the faults of feature models. BeTTy is an integrated techniques tool of the above two FAMA and SPLOT tools which use in software product line engineering to analyze the feature models. BeTTy is popular because it can generate random and high customized feature model, it enables to generate basic as well as extended attribute feature model. For functional testing (fault detection) it generates automated test data, and have an algorithm that supports user-defined optimization criteria for pessimistic cases (typical feature model) [12].

III. FEATURE MODEL ANALYSIS

Feature model has the different type of features given by the product designer, and that type of engineering known as domain engineering. In this section, we generate a model with some features and then analyze that same model on different tools which describe in the previous section and check the product configuration of the model.

A. Feature IDE

It is an open source tool where we design a feature model of "**Helloworld"** (root feature) with different features as shown in the figure. All the other features connect with root using mandatory and optional legends. These legends shows presence of features in the model, also fix the cardinal between features (**Feature** with **the world**). Feature IDE consists numerous of composers in which any feature model could be designed, the given model composed of feature IDE AHEAD compose.

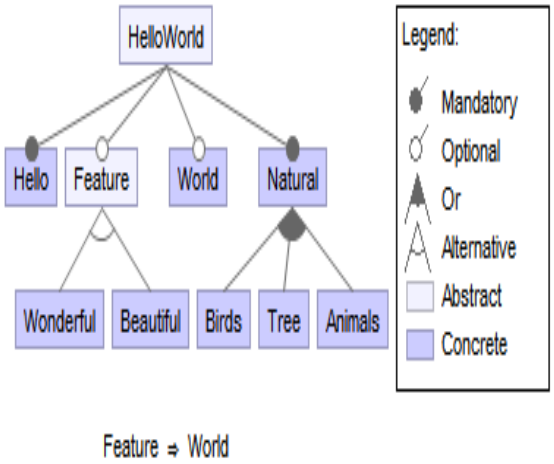

Figure.1

B. S.P.L.O.T

SPLOT is an online analytical tool [4] for feature model where we construct same feature model as shown in figure.2. SPLOT automatically analyze any given model and provide all the information like model statistics, analysis table, debugging table, and can also analyze the model with less and more feature.

Figure.2 define the automated analyses i.e. set of features in the model, also present which features are optional, mandatory, alternates, how many features are grouped together, cross-tree constraints (CTC) also their density type. Valid configuration, core features, and dead features are defined in model analysis.

SPLOT has default model composer SAT solver (Analyze Model debugging) describe the dead, common features and running time to analyze a model whereas BDD engine (Create Metrics) presents the number of configuration, variability degree, and other related information.

Debugging Analyses (Click to Run the SAT solver)	
Consistency	consistent
Running Time (ms)	0
#Dead Features	0
- Running Time (ms)	0
#Common Features	3 view
- Running Time (ms)	1

Metrics (Click to run the BDD engine)	
Count Configurations	35
- Running Time (ms)	7
Variability Degree (%)	3.418E0
- Running Time (ms)	7
#BDD Nodes	13
#BDD Variable Order Heuristic	Pre-CL-MinSpan

Figure.2

To simplify configuration process we check our model with full features and with fewer features. The given figure show decisions and propagations with SAT solver. Analyze on less and more feature propagation, SAT checks and SAT time in auto-compilation results (see config.1, 2) figure.3. But analyze a good difference in auto-completion and ordinary configuration (see config.1, 2 and 3) figure.3.

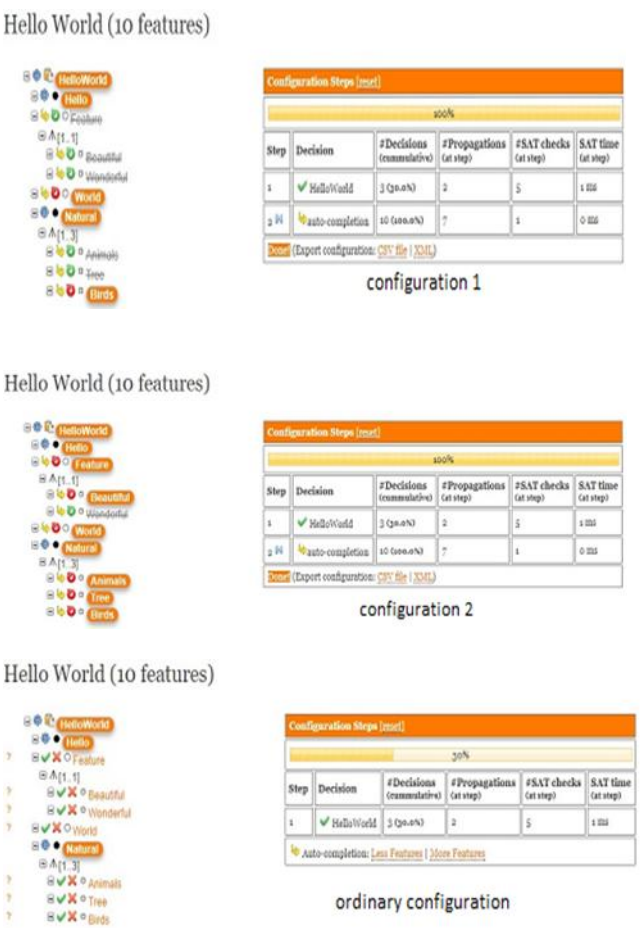

Figure.3

C. FaMa

FaMa is an offline tool [9] which run on the command prompt, FaMa uses further than analyzing software product lines strictly and can analyze any system that can be expressed as a set of features on a hierarchy, with or without attributes.

The FaMa tool can be used in four ways. It provides a shell front-end to final users, and three ways to integrate any designed application with FaMa: a java standalone version, a SOAP/WSDL web service, and a set of OSGi bundles.

FaMa shell: is the current front end for final users. A command-line interface where you can load models, and invoke analysis operations.

FaMa Web Service: provide a WSDL with most of FaMa operations.

FaMa OSGi: FaMa is also available as a set of OSGi bundles. OSGi (www.osgi.org) is a java specification for services integration.

FaMa standalone: There is an extensible library available in FaMa.

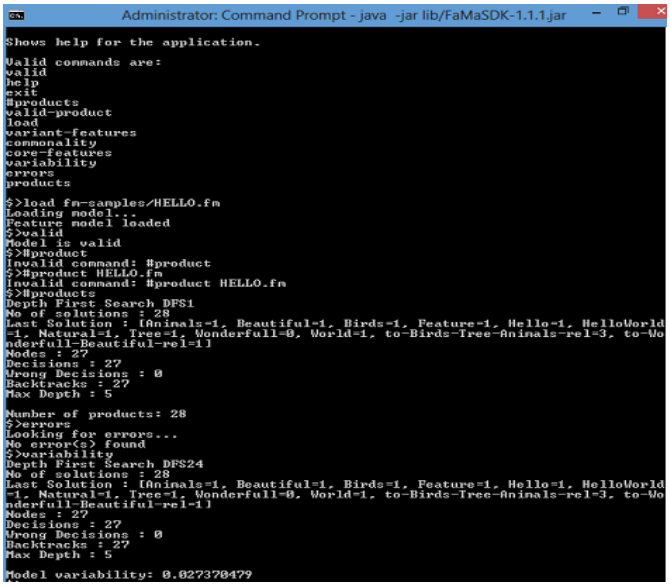

Figure.4.1

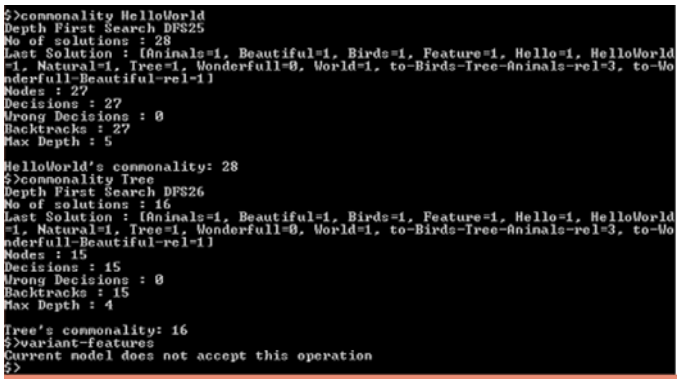

Figure 4.2

To analyze a designed model on FaMa shell which is a part of FaMa tool and analyze different criteria with the following commands by FaMa user manual:

Load: to load designed model.
Valid: to check the configuration of the model is valid or not.
Products: to check how many different products can be made by the features.
Errors: to analyze, if there is any error in the model.
Variability: to calculate the variability of the model
Commonality: to analyze a feature commonality in the model with other features.

Commands like **"load", "valid", "products", "errors", "variability"**" and their respective results represent in figure.4.1. Commands like "**commonality"**, **"variant–features", "core–features"** in figure.4.2, the commonality command required a feature name to show the commonality of that feature in the model.

D. BeTTY

BeTTy is a web-based tool, coded in Java language, and released in the jar file under license of LPGL3 can be downloaded from www.isa.us.es/betty, BeTTy website. BeTTy tool allows to generate feature models randomly, and that generation accomplished in two phases. Firstly, it generates a tree of features with the algorithm described by Thum et al.[11] secondly, to fulfill the requirements of a model it adds CTC (Cross-Tree Constraints) in the model. These requirements are, one constraint cannot be shareable by two features and, another one is no parent feature relation could not be part of constraints. A user can generate an output of the given feature model in different formats like SXFM, FaMaXML, FaMaTextFormate. DOT etc. as shown in figure.5.
BeTTy is a web-based online tool which environment seems like figure. After complete required entries user can fill the details of model want to be created like a number of model, features and cross-tree constraints percentage. Also, a user can fill the optional details like CTC percentage, the percentage of relationship (mandatory, alternative, maximum branching factor, the maximum number of sub-features in sets optionally. A user can generate only valid models by the select satisfiability optional.

IV. Analysis of Tools

The given table shows the comparison between tools on the basis of factors as availability of tools (FeatureIDE and FaMa) is offline i.e download and can run on the computer system, whereas SPLOT and BeTTy work online in which user fill required information about the model and got output. Model composer through which user can compose their model FeatureIDE have eight model composer, SPLOT only two, BeTTy produce automatic configuration, whereas in FaMa no composer available.
A model can have some faults, present tools FeatureIDE, FaMa, and BeTTy have the capability to detect faults. In A model is valid or not, or model has any bed feature or not detect by testing of model FeatureIDE and SPLOT test model and provide the output, on the other side FaMa just check that given model is valid or not.

Table 1. Comparision of tools on the basis of different parameters

Tools / Factors	FeatureIDE	SPLOT	FaMa	BeTTy
Availability of Tool	Offline	Online	Offline	Online
CTC(Cross-tree Constraint)	NA	Yes	Yes	Yes
Support Testing	Yes	Yes	Partially	NA
Fault Detection	Yes	NA	Yes	Yes
Products Generation	Yes	Yes	Yes	Yes
Statistics of model	Yes	Yes	NA	NA
Model Composer	Yes	Yes	NA	NA

V. Conclusion and Future Work

In this paper, we analyze four tools FeatureIDE, SPLOT, Fama and BeTTy in which two are web-based and other two are offline. Paper based on analysis tools with the using feature model. For which, we designed a feature model and check that model on above tools. After analyzing tools for designed model on the factors (Availability of Tool, Cross Tree Constraint, Support Testing, Fault Detection, Product Generation, Statistics of Model and Model Composer). At last we create a table to recognize that which factor supported by the tool, and which are not. There are numerous of tools and plugins which are available online or offline to analyze feature model on the same and other factors. In future, we intended to analyze some other tools on different factors including same factors we use here.

References

[1] K. Satendra, Rajkumar, "Test Case Prioritization Techniques for Software Product Line: A Survey" International Conference on Computing, Communication, and Automation (ICCCA), 2016.
[2] M. Qaiser, S. Muhammad "Software Product Line: Survey of Tools", Dept. of Computer and Information Science, Linköping University, 2010.
[3] C. Paul, N. Linda. "Software product lines: Practices and patterns", Addison-Wesley, 2001.
[4] S.P.L.O.T.: Software Product Lines Online Tools.http://www.splot-research.org.
[5] S. Segura. "Functional and Performance Testing of Feature Model Analysis Tools Extending the FaMa
Ecosystem", Ph.D. thesis, Dept. of Computer Languages and Systems, University of Seville, 2011.
[6] T. Thomas, M. Jens, FeatureIDE: Overview October 23, 2015.

[7] B. David, S. Sergio, T. Pablo R.C. Antonio, "FAMA: Tooling a Framework for the Automated Analysis of Feature Models".
[8] B. Don, B. David, R.C. Antino, "Automated analysis of feature models: Challenges ahead", Communications of the ACM, December 2006.
[9] FaMa Tool Suite. http://www.isa.us.es/fama.
[10] Mendonça, M., Branco, M., and Cowan, D. (2009). "S.p.l.o.t.: Software product lines online tools". In 24th Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA), pages 761–762.
[11] T.Thomas, B.Don, K. Christian, "Reasoning about edits to feature models", In International Conference
on Software Engineering, pages 254-264, 2009.
[12] S.Sergio, G. José A., B. David, P. José A. and R.C. Antonio, "BeTTy: Benchmarking and Testing on the Automated Analysis of Feature Models", VAMOS '12, January 25-27, 2012 Leipzig, Germany.
[13] S. Kumar and Rajkumar, "Cost-Based Test Case Prioritization Technique for Software Product Line," *Int. J. Sci. Prog. Res.*, vol. 40, no. 115, 2017.

Figure 5

Authors Profile

Mr.Ashish Saini pursued Bachelor of Science from CCS University, Meerut. He pursued MCA from Uttarakhand Technical University, Dehradun. He is currently pursuing Ph.D. at Department of Computer Science in Gurukula Kangri Vishwavidyalaya, Haridwar. He published more than 5 papers in various international and national conferences and it's also available online. His research work areas are Software Testing, Software Product Line, Feature Model, and Cloud Computing.

Dr.Rajkumar pursued Bachelor of Science and Master of Science from Gurukula Kangri Vishwavidyalaya, Haridwar. MCA from IGNOU. He completed his Ph.D. in Wireless Sensor network from Gurukula Kangri Vishwavidyalaya, Haridwar. He has Published more than 15 research in different areas in the reputed journal (SCI, Scopus Indexed) and conferences including IEEE. He has more than 20 years of teaching experience and more than 10 years experience in research. His area of research in the wireless sensor network, interconnection network, software product line, cloud computing, software testing.

Mr.Satendra Kumar pursued Bachelor of Technology from MJP Rohilkhand University Bareilly in the year 2008 and Master of Technology from YMCA University of Science and Technology Faridabad, Haryana. He is currently pursuing Ph.D. at Department of Computer Science in Gurukula Kangri Vishwavidyalaya, Haridwar. He is a lifetime member of ISRD Journal. He published more than 8 papers in various reputed journals and conferences including IEEE and it's also available online. His research work areas are Software Testing, Software Product Line, Feature Model Analysis, Clustering and Web Minning.

