LATIN.SCIENCE XVI Congresso Latino-americano de Software Livre e Tecnologias Abertas LATINOWARE 2019

Open Source Software on the Research of Extractive Adoption of Software Product Lines

Daniele Wolfart
PPGComp - Western Paraná State University
Cascavel, Brazil
danielewolfart@gmail.com

Wesley Klewerton Guez Assunção
COTSI - Federal University of Technology - Paraná
Toledo, Brazil
wesleyk@utfpr.edu.br

Jabier Martinez
Tecnalia
Derio, Spain
jabier.martinez@tecnalia.com

***Abstract*—Beyond the main purpose of Open Source Software (OSS) to provide open industrial and personal solutions, pieces of OSS are also the subject of many research efforts. In this work we focus on OSS usage as case study subjects in the context of extractive adoption of software product lines. This research field is related to re-engineering existing system variants towards a more systematic reuse for the creation and management of a family of products. By analyzing a catalog of case studies, we provide an overview and a discussion of the current research state-of-the-practice of OSS usage across the different phases of the re-engineering process. We complement this work with the identification of available OSS tools to support this process to show, as conclusion, the healthy contribution that OSS communities are directly or indirectly making to this active research field.**

***Keywords*-Open Source Software; Research; Software Product Lines; Extractive adoption of software product lines; re-engineering**

I. Introduction

Open source software (OSS) is an open development model where software is collectively created, used, and improved. In an OSS, artifacts are publicly accessible and anyone can copy and modify them, or add new features, to fulfill his/her requirements [1]. OSS has been receiving great attention from practitioners, researchers, academia, and both public and private companies [2]. The effort of Open source communities is paramount for many companies, being OSS the basis for their products, services, and/or internal operation while keeping (or even increasing) their innovative and competitive levels [3].

It is also the case that OSS provides benefits in the context of research and education as this ecosystem of collaboration helps to make progress on research and educational challenges [4], [5]. For instance, systems and software under OSS licenses are exhaustively used as case studies to evaluate or compare new approaches. Also, the tool-support for these approaches can be OSS as well, enabling the community to enhance or improve the results. Compared to confidential software and private software, OSS usage seems the most appropriate way towards reproducible research.

The goal of this work is to present the use of OSS on an specific research field in software engineering: The extractive adoption of Software Product Lines (SPLs) from existing system variants. An SPL is a systematic reuse approach where a set of software products (a family of products) share common parts and cover the variability needs of a specific domain [6]. The most common way in which companies adopt SPLs is by re-engineering existing systems [7], known as the extractive approach [8] for SPL implementation.

The main contribution of this work is to make the software engineering community aware on how OSS has been used to support research on extractive adoption of SPLs. In addition, we want to motivate authors to make their case studies and tools available as OSS or build on top of them.

II. Background

A. Extractive adoption of SPLs

The extractive adoption of SPLs is performed by a re-engineering process. Re-engineering is "*the examination and alteration of a subject system to reconstitute it in a new form and the subsequent implementation of the new form*" [9]. An overview of the reengineering process with focus on adoption of SPLs is illustrated in Figure 1. On the left side we can find the existing system variants where the solid line represents the entire process of re-engineering, which is usually composed of different phases, presented with dashed lines.

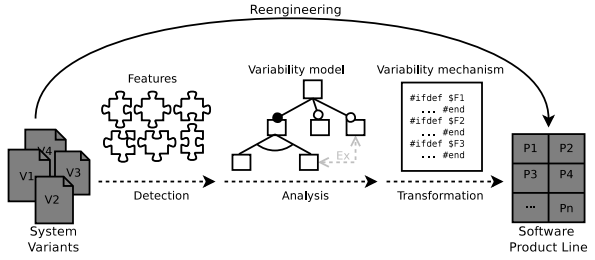

Figure 1. The re-engineering process from a set of variants to a software product line and its intermediate phases [7]

The generic phases of the re-engineering process of existing systems into SPLs are:

(i) *Detection phase* is the beginning of the process, devoted to detect the variabilities and commonalities among existing products. The variabilities and commonalities are represented in terms of features. Common support in this phase is given by feature identification and location techniques, which aim at locating the elements responsible for implementing the system functionalities. The management of the features and the mapping/traceability to the software artifacts that implement them are tasks of the SPLE domain engineering process;

(ii) *Analysis phase* involves the organization of discovered variabilities and commonalities. This step is devoted to create the variability model, shown in the middle of Figure 1, to express the valid combinations of features of an SPL. Feature Models (FMs) are the most popular form of variability model; and

(iii) in the *Transformation phase* the artifacts that implement the features and the variability model are used to create the SPL, using a variability mechanism. For instance, the simplest mechanism is based on `#ifdef` source code annotations that are pre-processed (i.e., removing code fragments when a given feature is not selected in a configuration). Illustrative annotations are shown on the right side of Figure 1.

B. ESPLA: A catalog of cases studies

The source of information for our work is the ESPLA (Extractive Software Product Line Adoption) catalog of case studies[1] [10]. The aim of this open catalog is to foster the advance of this field by providing comprehensive and accessible information about case studies scattered in the literature. ESPLA is designed to be a collaborative catalog maintained by the SPL research community. Currently the catalog is composed of 130 case studies[2]. The catalog shows a great diversity in terms of case study origin, domain, type of artifact, and size.

The main pieces of information in the catalog are:

- The name of the case study and a brief description;
- References in the literature using it and the reference of the original publication presenting or using the case study;
- List of websites for direct download of the case study or for getting more information;
- Whether the variants are publicly available and whether the origin is industrial, open-source, academic, illustrative or others;
- The type of artifact (e.g., source code, models, requirements, documentation) and a more detailed description of the type of artifact;
- The number of variants and the average, minimum and maximum size of the variants;
- The activities of extractive SPL adoption used with the case study from the literature and the strategies already used to tackle it;
- A description about how the variants were created;
- The number of features and the availability of the variability model and the name of the features.

[1]https://but4reuse.github.io/espla_catalog/

[2]Last update on May 20th, 2019

III. Analysis of the ESPLA catalog from an OSS perspective

In the following we describe, analyze and discuss our findings about the OSS case studies. Regarding the origin of the case studies, Figure 2 presents a pie chart with the amount of case studies in each category. We can observe that the most common case studies are from industry (45) and OSS projects (44). The 44 OSS projects available in the ESPLA catalog are used in 72 research papers. This shows us how representative is OSS to support research on extractive adoption of SPLs.

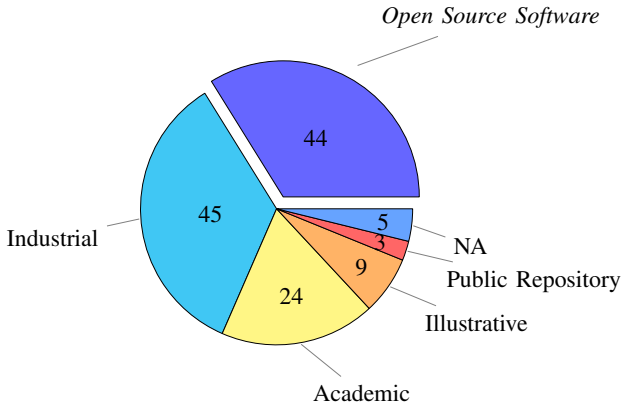

Figure 2. Origin of the case studies in the ESPLA catalog [10] showing the large amount of case studies that consider OSS as artifact/data set

Considering the goal of our work, we focus the following analyses of the catalog considering exclusively the 44 OSS case studies. Figure 3 presents the number of those case studies categorized by the type of artifacts.

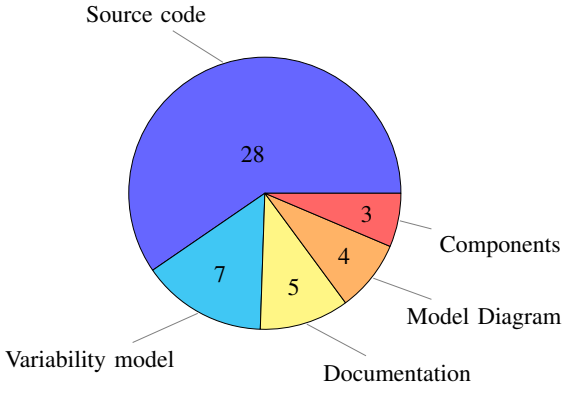

Figure 3. Number of OSS case studies by type of artifact

The most common artifact is the source code, present in 63% (28) of the OSS projects. The programming languages of the source code are Java, JavaScript, and C. However, other types of artifacts produced during the software development cycle are also covered, such as documentation, variability models, models/diagrams, and components. Some case studies have more than one type of artifact. Researches can benefit from this range of different types of publicly available artifacts to evaluate their proposed approaches.

Given that the input for the re-engineering process is a set of system variants, we considered in our analysis the number of variants in each OSS project. As presented in Figure 4, we can see that most of the case studies are composed of less than five variants. It is almost impossible to generalize but this could lead to infer that this number might be a common scenario in practice. However, there are case studies with more than 51 variants, allowing researches to evaluate the scalability of their approaches.

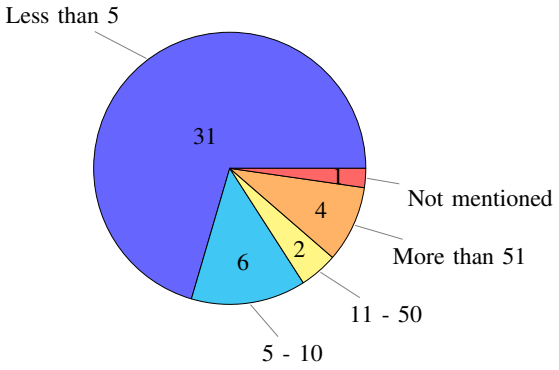

Figure 4. Number of OSS case studies by number of variants

Figure 5 presents the number of OSS projects considering the number of features. We can observe a good distribution in terms of feature number diversity. Most of them have between 10 and 50. Considering case studies with the number of features in the category of 10-50 together with the category 51-100, we have a total of 22 OSS projects, which 20 of them are based on source code. Unfortunately, many case studies do not mention the number of features.

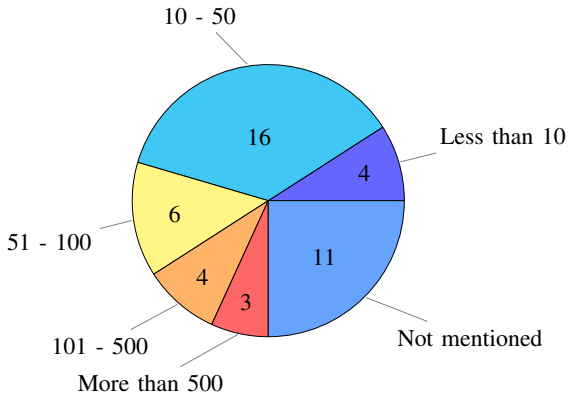

Figure 5. Number of OSS case studies by number of features

To analyze how the OSS case studies are used for covering the re-engineering process, Figure 6 shows a Venn diagram with the number of case studies in the phases of the process. We can see that the greater number of case studies (13) are used in all phases of the process. 12 case studies are used only in the analysis phase, 10 in detection and analysis, and 8 only in detection. The greater number of case studies used for all the phases of the re-engineering process is surprising. A previous work that considered all types of case studies (independently if they were OSS or not) concluded that most of them were used only for detection and analysis [7]. It seems that, because of the easy access to all artifacts of OSS projects, it allows to deal with all phases of the re-engineering process.

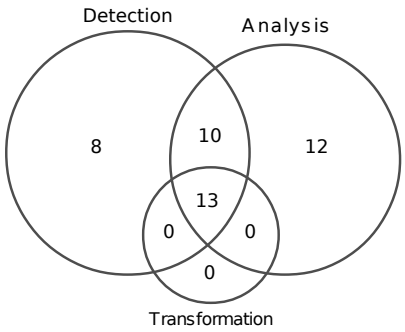

Figure 6. Number of OSS projects per re-engineering phase

Regarding the activities related to the re-engineering process, Table I presents the number of OSS projects used in these activities. Feature model synthesis, which is a part of the analysis phase, is the most covered activity. This was expected as a feature model is an important initial artifact to plan and construct SPLs. The activities of feature identification and location, part of the detection phase, are the second and third most common covered activities, responsible for identifying and tracing the features to their implementation. The phase of transformation is also represented, namely by the activity of reusable assets construction, reusable assets extraction, and decomposition of artifacts.

Table I
TABLE OF ACTIVITIES COVERED BY OSS CASE STUDIES

Activities	**# OSS case studies**
Feature Model Synthesis	27
Feature identification	24
Feature location	18
Reusable assets construction	8
Feature constraints discovery	7
Reusable assets extraction	4
Decomposition of artifacts	1
Assessment of reuse potential	1
Feature Model extraction	1
Traceability recovery	1
Reference architecture discovery	1
Feature naming	1
N/A	1

We also carried out an analysis of which types of artifacts are used in each phase of the re-engineering process. We observed that source code is commonly used in the three phases of the process. The phase of analysis deals with variability models. When considering detection phase, source code, models/diagrams, and documentation are the types of artifacts took into account. Components are mostly used in the Feature Location. The only activity that deal with all types of artifacts is feature model synthesis, part of the analysis phase.

IV. OSS TOOL-SUPPORT FOR EXTRACTIVE SPL ADOPTION AND GENERAL DISCUSSION

Besides OSS as case study subjects, this section discuss other dimensions of OSS in this research domain. The most relevant point is to present OSS tool-support for the re-engineering activities. We took into account the mapping study on re-engineering systems variants into SPLs, where the authors identify and describe 19 tools [7]. From this set of tools, we identified 6 OSS tools to support extractive adoption of SPLs. They are: ETHOM[3], CloneDifferentiator[4], SPLevo[5], BUT4Reuse[6], ExtractorPL[7], and ECCO[8]. They are all research tools with different levels of maturity but, overall, they cover all the re-engineering process.

As general discussion, it is important to mention that re-engineering and SPL adoption is not as frequently covered as other topics in SPL pedagogical material [11]. The availability of OSS tools and case studies can help to fill this gap for this relevant topic in SPL engineering education.

In Section III we analysed the case studies. Some case studies are transformed or serve as benchmarks for activities in extractive SPL adoption. The use of common benchmarks (established ground-truth and evaluation criteria) can foster a research field through comparable results of different techniques. Several benchmarks were identified in the SPL field covering different evolution scenarios of variant-rich systems [12] including those related to the re-engineering process. For these scenarios, OSS case studies transformed into well-defined benchmarks play a key and almost exclusive role. However, some gaps were identified in terms of diversity and missing characteristics of the benchmarks.

V. CONCLUSION

This work presents an analysis of the extractive Software Product Line adoption field from the perspective of OSS usage. We have shown evidences that OSS is a major source of case study subjects, and the results of more detailed analyses show that they have enough diversity to fit the evaluation needs of several phases and activities of the re-engineering process. Besides OSS case studies, we also presented OSS tools to support this process. As future work, the gaps identified in education and in benchmarking activities will be tackled.

[3]http://www.lsi.us.es/~dbc/material/ssbse2012/

[4]https://sites.google.com/site/yinxingxue/home/projects/clonedifferentiator

[5]https://github.com/kopl/SPLevo

[6]https://but4reuse.github.io/

[7]https://pages.lip6.fr/Tewfik.Ziadi/sac14/

[8]http://jku-isse.github.io/ecco/

ACKNOWLEDGMENT

This work was supported by the Brazilian National Council for Scientific and Technological Development (CNPq), grant 408356/2018-9.

REFERENCES

[1] A. Fuggetta, "Open source software—an evaluation," *Journal of Systems and Software*, vol. 66, no. 1, pp. 77 – 90, 2003.

[2] S. Dhir and S. Dhir, "Adoption of open-source software versus proprietary software: An exploratory study," *Strategic Change*, vol. 26, no. 4, pp. 363–371, 2017.

[3] H. Munir, K. Wnuk, and P. Runeson, "Open innovation in software engineering: a systematic mapping study," *Empirical Software Engineering*, vol. 21, no. 2, pp. 684–723, Apr 2016.

[4] J. Bishop, C. Jensen, W. Scacchi, and A. Smith, "How to use open source software in education," in *47th ACM Technical Symposium on Computing Science Education*, ser. SIGCSE '16. New York, NY, USA: ACM, 2016, pp. 321–322.

[5] G. Von Krogh and E. Von Hippel, "The promise of research on open source software," *Management science*, vol. 52, no. 7, pp. 975–983, 2006.

[6] F. J. v. d. Linden, K. Schmid, and E. Rommes, *Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

[7] W. K. G. Assunção, R. E. Lopez-Herrejon, L. Linsbauer, S. R. Vergilio, and A. Egyed, "Reengineering legacy applications into software product lines: a systematic mapping," *Empirical Software Engineering*, vol. 22, no. 6, pp. 2972–3016, Dec 2017.

[8] C. W. Krueger, "Software reuse," *ACM Computing Surveys*, vol. 24, no. 2, pp. 131–183, Jun. 1992.

[9] E. Chikofsky and I. Cross, J.H., "Reverse engineering and design recovery: a taxonomy," *IEEE Software*, vol. 7, no. 1, pp. 13–17, 1990.

[10] J. Martinez, W. K. G. Assunção, and T. Ziadi, "ESPLA: A Catalog of Extractive SPL Adoption Case Studies," in *21st International Systems and Software Product Line Conference - Volume B*, ser. SPLC '17. New York, NY, USA: ACM, 2017, pp. 38–41.

[11] M. Acher, R. E. Lopez-Herrejon, and R. Rabiser, "Teaching software product lines: A snapshot of current practices and challenges," *TOCE*, vol. 18, no. 1, pp. 2:1–2:31, 2017.

[12] D. Strüber, M. Mukelabai, J. Krüger, S. Fischer, L. Linsbauer, J. Martinez, and T. Berger, "Facing the truth: Benchmarking the techniques for the evolution of variant-rich systems," in *23rd International Systems and Software Product Line Conference*, Paris, France, 2019.