How I Met Your Implemented Variability: Identification in Object-Oriented Systems with symfinder

Johann Mortara
Université Côte d'Azur, CNRS, I3S
Sophia Antipolis, France
johann.mortara@univ-cotedazur.fr

Philippe Collet
Université Côte d'Azur, CNRS, I3S
Sophia Antipolis, France
philippe.collet@univ-cotedazur.fr

ABSTRACT

Variability-rich object-oriented systems are often not organized as fully-fledged software product lines, and implement their variability in a single code base using the mechanisms provided by the supporting language (*e.g.*, inheritance overloading, design patterns). This makes variability identification and management very difficult. In this half-day tutorial open to both academics and industrials, we present how the *symfinder* toolchain can help one to better understand how variability is implemented in a single codebase Java system, relying solely on a specific code analysis and an adapted visualization. After presenting the underlying concepts on which *symfinder* is based (*i.e.*, symmetries in code, density), the participants will be able to use the toolchain and visualize the potential variation points and variants identified by *symfinder* in their own projects or in provided large-scale open-source projects.

CCS CONCEPTS

• **Software and its engineering** → **Software product lines**; **Software maintenance tools**; *Object oriented architectures.*

KEYWORDS

symfinder, variability, variability identification, visualization

Motivation. In variability-rich object-oriented systems, variability can often be implemented in a single code base using the mechanisms provided by the supporting language (inheritance, overloading, design patterns…). However, such systems are often not developed as product lines, which leads to a misalignment between domain features, and the implemented variation points (*vp*-s) and variants, finally making their identification and management difficult. Moreover, the lack of an explicit set of features or of code variants to be differentiated impedes the applicability of feature location techniques [3]. There are many cases in which one can only rely on the code assets to identify variability implementations.

symfinder is a toolchain whose goal is to assist in this activity [1]. It relies on the notion of symmetry in object-oriented constructs [5] to automatically identify potential *vp*-s and variants in a single code base, and provides a web-based graph visualization of the identified variability implementations that shows dense zones of potential *vp*-s and variants to the user. The event will be organized as follow:

Part 1 (theoretical and interactive) – Motivations & introduction to *symfinder*. The tutorial will start with a presentation of the challenges related to feature location and feature identification [3], and how these problems impact both academic and industrial communities. We will then present the *symfinder* toolchain and how it makes use of the symmetry in object-oriented constructs to identify the variability implementations in a single object-oriented code base, relying on visualizations and findings on previous work [2, 4].

Part 2 (theoretical and practical) – First contact with *symfinder*. In a second part, participants will be invited to use *symfinder*. After a quick technical presentation of the toolchain, attendees will be walked through by executing an already set up experiment step by step, from execution to interpretation of the final results.

Part 3 (practical) – Guided use of *symfinder*. In a third part, attendees will have the opportunity to set up *symfinder* to analyze their own project(s), or provided large open-source subject systems. Visualizations obtained by the attendees will be uploaded in a *hall of fame* on *symfinder*'s website.

Part 4 (interactive) – Exchange time. Finally, all attendees will have the possibility to share their findings on the different studied systems, exchanging on the form of variability architectures identified, and on their experience with *symfinder*.

Link to material. All materials are present on the tutorial's page: https://deathstar3.github.io/SPLC2021-symfinder-tutorial.

REFERENCES

[1] Johann Mortara, Xhevahire Tërnava, and Philippe Collet. 2019. symfinder: A Toolchain for the Identification and Visualization of Object-Oriented Variability Implementations. In *the 23rd International Systems and Software Product Line Conference.*

[2] Johann Mortara, Xhevahire Tërnava, and Philippe Collet. 2020. Mapping Features to Automatically Identified Object-Oriented Variability Implementations - The case of ArgoUML-SPL. In *14th International Working Conference on Variability Modelling of Software-Intensive Systems (VaMoS '20).*

[3] Julia Rubin and Marsha Chechik. 2013. A survey of feature location techniques. In *Domain Engineering.* Springer, 29–58.

[4] Xhevahire Tërnava, Johann Mortara, and Philippe Collet. 2019. Identifying and visualizing variability in object-oriented variability-rich systems. In *the 23rd International Systems and Software Product Line Conference.*

[5] Liping Zhao and James Coplien. 2003. Understanding symmetry in object-oriented languages. *Journal of Object Technology* 2, 5 (2003), 123–134.