FeatRacer: Locating Features Through Assisted Traceability (Summary)

(article starts on next page)

FeatRacer: Locating Features Through Assisted Traceability (Summary)

Mukelabai Mukelabai [1], Kevin Hermann [2], Thorsten Berger [3], and Jan-Philipp Steghöfer [4]

Abstract: This extended abstract summarizes our paper with the same title published at the IEEE Transactions on Software Engineering Journal (TSE) 2023 [Mu23].

**Keywords:** feature location, traceability, recommender

1 Introduction

Feature Location is one of the most common tasks performed by developers when developing software systems. Most commonly it is done during the maintenance and evolution of software systems, as developers must identify the exact locations in a codebase where specific features are implemented. Unfortunately, it is a laborious and error-prone task, since the knowledge of features and their location fades, the developers working on a project change, and features often scatter across the whole codebase.

To this end, many *automated feature location techniques* have been proposed, which aim to retroactively recover features.Unfortunately, such approaches are not useful in practice since they recover only coarse-grained locations, produce too many false positives, and require large training datasets. Alternatively, *recording features during development*, when their location is still in a developer's mind, circumvents these issues. Still, developers easily forget to annotate code and it is also a costly process, especially during software evolution as these recordings require updates.

In this work, we take a different approach in addressing the *feature location problem* (a.k.a., *concern location* or *concept assignment problem*). We present FeatRacer, which addresses the shortcomings of both feature recording and automated feature location by allowing developers to record features proactively and continuously during development. FeatRacer relies on embedded code annotations and a machine-learning-based recommender system. FeatRacer learn the developer's feature recording practices within the project and reminds

1 University of Zambia, Zambia, and Chalmers | University of Gothenburg, Sweden,
mukelabai.mukelabai@unza.zm, ⓘ https://orcid.org/0000-0002-3868-4319

2 Ruhr University Bochum, Germany,
kevin.hermann@rub.de, ⓘ https://orcid.org/0009-0004-6207-4045

3 Ruhr University Bochum, Germany, and Chalmers | University of Gothenburg, Sweden,
thorsten.berger@rub.de, ⓘ https://orcid.org/0000-0002-3870-5167

4 XITASO Gmbh IT & Software Solutions, Germany, and Chalmers | University of Gothenburg, Sweden,
jan-philipp.steghoefer@xitaso.com, ⓘ https://orcid.org/0000-0003-1694-0972

the developer about potentially missing features, when they forget to annotate code. This makes it easier for developers to recover the fine-grained locations of features.

Assets can be annotated on three levels of granularity levels: folder, file and code fragment We engineered metrics that measure characteristics of features, which are utilized to estimate the relatedness of assets that implement a given feature. While engineering these metrics, we found that data considering process-related metrics such as contributions made by specific developers show a higher correlation with the prediction performance of FeatRacer than commit-related metrics such as number of lines added. When FeatRacer finds a non-annotated asset during a revision, its machine-learning classifiers can predict which feature it implements based on the asset's metrics.

We show, that FeatRacer outperforms traditional automated feature location based on Latent Semantic Indexing (LSI) and Linear Discriminant Analysis (LDA)—which are two of the most common techniques used for feature location recovery—when predicting features for 4,650 commit changesets from the histories of 16 open-source projects which have been developed on average for three years between the years 1985 and 2015.

When comparing FeatRacer to traditional feature location techniques among all 16 projects, FeatRacer showed a 3x higher precision and a 4.5x higher recall, with an average precision and recall of 89.6 %. We show, that FeatRacer is already effective in predicting feature locations small datasets, by running it on the first five commits of our subject projects. In contrast to traditional feature location techniques, for which developers need to investigate entire project histories, or need a lot of effort and time from the developer to write complex search queries for features, FeatRacer only requires on average 1.9ms to effectively learn from a project's past code fragments, and only 0.002ms to recommend appropriate feature annotations in non-annotated code.

2   Data Availability

An online appendix with our used datasets, implementation of LSI and LDA, and evaluation data can be found online [Th23].

Bibliography

[Mu23] Mukelabai, Mukelabai; Hermann, Kevin; Berger, Thorsten; Steghöfer, Jan-Philipp: FeatRacer: Locating Features Through Assisted Traceability. IEEE Transactions on Software Engineering, pp. 1–23, 2023.

[Th23] The Authors: Online Appendix. `https://bitbucket.org/easelab/featracer/`, 2023.