How to Extract “Feature Architecture” of your Software

Tajmilur Rahman, PhD

Feature Architecture [1] or Feature-toggle Architecture is an architecture which represents the cross-cut between software features and the concrete modules at the same time. The feature toggle architecture can be extracted from source code and shows which features affect which modules and vice versa. Feature toggles can be named as ‘flags’, ‘feature flips’, or ‘switches’ that cover the code that implements a feature. It’s a very old technique used by software developers. They are used by large web companies to quickly enable and disable a feature in production [2]. Using the directory-to-module mapping derived as part of the concrete architecture, toggles can be extracted contained in each module. Toggles that are contained in multiple modules indicate a feature relationship between the modules. The feature dependencies do not necessarily indicate modular architectural violations as features must span multiple modules. For example, the “EnableAutoFill” feature toggle in Google Chrome must be present in the UI module to control some code to allow the user to select the content to autofill and also be present in the Data Persistence module to store potential autofill content. Our novel contribution of extracting the feature toggle architecture allows researchers and developers to view the features that are present in a module and the modules that are necessary for a feature.

In this tutorial, we will learn how we can extract feature architecture of a software application. Why extracting feature architecture is important, and when this method is applicable.

References

[1] M. T. Rahman, Rigby, P. C., & Shihab, E. (2019). The modular and feature toggle architectures of Google Chrome. *Empirical Software Engineering*, *24*(2), 826-853.

[2] M. T. Rahman, Querel, L. P., Rigby, P. C., & Adams, B. (2016, May). Feature toggles: practitioner practices and a case study. In *Proceedings of the 13th international conference on mining software repositories* (pp. 201-211).