

# Einstiegspunkte für Design und Codierung bei einer Wartungsaufgabe oder Fehlermeldung

Felix Fröhlich und Thorben Wiese

Universität Hamburg  
Fakultät für Mathematik,  
Informatik und Naturwissenschaften  
Department Informatik

**Abstract.** Bei der Entwicklung und Wartung von Software spielt die Identifikation des Nutzen von Programmabschnitten eine große Rolle, um entsprechende Funktionen einer Software zu ändern oder zu reparieren. Diese *Feature Locations* stellen einen Einstiegspunkt in den Quelltext für Design- oder Code-Änderungen dar und können mithilfe verschiedener Technologien ermittelt werden. In dieser Seminararbeit stellen wir unterschiedliche Technologien und deren Verfahren vor und geben einen Überblick über geeignete Nutzungsfelder.

## 1 Einleitung

Die erste Quelle [1].

...

Ziel dieser Seminararbeit ist es, die verschiedenen Analyseverfahren und Technologien zur Erkennung von Features im Code zu beschreiben und zu vergleichen.

## 2 Begriffe

Für die Vorstellung der Analyseverfahren sollen zunächst einige Begriffe definiert und erklärt werden.

### Feature

Ein Feature ist ein Software Artefakt, das eine spezifische Funktionalität implementiert [2]. Diese Funktionalität wird in natürlicher Sprache beschrieben und wird von einem Programmabschnitt wiedergespiegelt. Ein Feature besteht üblicherweise aus einem Namen, einer Bedeutung (Intension) und einer Erweiterung (Extension) [3].

### Feature Location

Der Prozess der Feature Location beschreibt die Identifikation der Beziehung zwischen Features und deren Implementierung. Dabei liegt die Beschreibung des Features in natürlicher Sprache vor, die dann einem entsprechenden Codeabschnitt zugeordnet werden soll (Mapping) [1].

### 3 Analyseverfahren

In diesem Abschnitt sollen verschiedene Analyseverfahren vorgestellt werden, die das Finden von Feature Locations ermöglichen.

#### Program Dependence Analysis (PDA)

#### Trace Analysis

#### Latent Semantic Indexing (LSI)

#### 3.1 Statische Feature Location Technologien

In diesem Abschnitt sollen statische Technologien zum Finden von Features im Quelltext vorgestellt werden.

#### Statische Analyse

Die Analyse von Quelltext zu einem Zeitpunkt, bei dem das Programm kompiliert wird und noch nicht ausgeführt wurde, wird statische Analyse genannt. Hierbei werden mithilfe von zum Beispiel Datenfluss-Analyse und Kontrollgraphen alle Abhängigkeiten und Funktionsaufrufe innerhalb des Codes analysiert und es können unter anderem Fehler wie zum Beispiel Race-Conditions oder Buffer-Overflows identifiziert werden. Dieser Prozess wird häufig von automatisierten Tools durchgeführt [4].

#### Technologie Beispiele

Robillard et al. [35]

Shao et al. [40]

#### 3.2 Dynamische Feature Location Technologien

#### Dynamische Analyse

Allgemein dynamische Analyse

#### Technologie Beispiele

Wong et al. [49]

Liu et al. [25] (SITIR)

#### 3.3 Textuelle Feature Location Technologien

Allgemein textuelle Analyse, Tools beschreiben, Beispiele

## **4 Vergleich**

## **5 Fazit**

Für welchen Zweck welches Analyseverfahren und welche Technologie Ergebnis wahrscheinlich: Alles gar nicht so schlecht, je nach Bedarf muss eine Technologie ausgewählt werden oder eventuell mit einer anderen kombiniert werden.

## Literaturverzeichnis

- [1] Rubin J, Chechik M. A Survey of Feature Location Techniques. University of Toronto, Department of Computer Science; 2013.
- [2] IEEE. Std. 829;. Available from: <https://standards.ieee.org/findstds/standard/829-2008.html>.
- [3] Chen K, Rajlich V. Case Study of Feature Location Using Dependence Graph. Wayne State University Department of Computer Science;. Available from: <http://www.cs.wayne.edu/~severe/publications/Chen.IWPC.2000.FeatureLocation.pdf>.
- [4] Open Web Application Security Project (OWASP). Static Code Analysis. Open Web Application Security Project (OWASP);. Available from: [https://www.owasp.org/index.php/Static\\_Code\\_Analysis](https://www.owasp.org/index.php/Static_Code_Analysis).