Technische Hochschule **Rosenheim**

Fakultät für Informatik

Studiengang Software- und Systems-Engineering

Erkennung von Design Patterns in Quellcode durch Machine Learning

Master Thesis

von

Mehmet Aslan

Datum der Abgabe: tt.mm.jjjj
Erstprüfer: Prof. Dr. Marcel Tilly
Zweitprüfer: Prof. Dr. Kai Höfig

Eigenständigkeitserklärung / Declaration of Originality

Hiermit bestätige ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken (dazu zählen auch Internetquellen) entnommen sind, wurden unter Angabe der Quelle kenntlich gemacht.

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Rosenheim, den tt.mm.jjjj

Vor- und Zuname

Kurzfassung

text

Schlagworte:

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Motivation

1.1 Einführung in Design Patterns

Entwurfsmuster oder auf Englisch 'Design Patterns' sind bewährte Lösungsansätze für wiederkehrende Probleme, die bei der Konzeption der Software-Architektur oder während der Implementierung der Software eingesetzt werden kann. Dabei dienen diese Entwurfsmuster als eine Art Blaupause, die es Software-Entwicklern ermöglicht, erprobte Lösungsstrategien für häufig auftretende Probleme in der Software-Entwicklung anzuwenden. Durch den Einsatz von etablierten Entwurfsmustern können Software-Entwickler für die Software bei korrekter Anwendung unter anderem erhöhte Wartbarkeit, Wiederverwendbarkeit von Komponenten, Verständlichkeit und Skalierbarkeit ermäglichen das widerrum in qualitativ besserer Software resultiert. Dabei sollte beachtet werden, dass Design Patterns als Vorlage zu betrachten sind. Je nach Einsatzgebiet muss die Anwendung des Entwurfsmusters evaluiert und für den konkreten Fall individualisiert werden. Deshalb existiert keine universelle anwendbare Iteration eines Design Patterns, die unabhängig von Anwendungskontext eingesetzt werden kann. Dies resultiert in varierenden Anwendung von Entwurfsmustern abhängig von jeweiligen Einsatzgebiet. Im weiteren Entwicklingszyklus der Software werden durch neue oder geänderete Anforderungen bereits eingesetzte Implemtierungen von Entwurfsmustern modifiziert, entfernt oder neue werden hinzugefügt. Währenddessen besteht die Gelegenheit, dass durch mangelnder Dokumentation oder anderer Gründe die Entscheidungen, weshalb Entwurfsmuster so eingesetzt sind wie es eingesetzt werden, verloren gehen. Dadurch besteht die Gefahr, dass angewendete Design Patterns im weiteren Verlauf derer Entwicklung nicht mehr wiederzuerkennen sind. Aus diesem Grund ist die Etablierung eines Prozesses von Vorteil, das in der Lage ist, Implementierungen von Entwurfsmustern aus einem Software-System zu extrahieren und dieses konkret benennen. Vorallem der Einsatz von Maschine Learning für die Klasszifierung ist hier vorteilhaft, wodurch das Potenzial besteht, vorher nicht gesehene Implementierung von Design Patterns zu erkennen. Durch solch einen Prozess können durch die Erkennung von eingesetzten Entwurfsmustern auf konkrete und verlorenengegangene Design-Entscheidungen zurückgeschlossen werden, welche zukünftige Design-Entscheidungen für das Software-System beeinflussen können. Der Fokus dieser Arbeit besteht daran, solch ein Prozess zu etablieren, welches für eingebenes Set von Quellcode-Dateien diese mit Hilfe von Maschine Learning ein potenziellen Entwurfsmuster zuzuteilen.

1.2 Untersuchungsfragen

Das Ziel dieser Arbeit besteht aus der Etablierung eines Prozesses, womit durch Einsatz von Maschine Learning für ein Set von Quellcode-Dateien ein Design Pattern zuzuordenen. Um solch ein Prozess zu entwicklen, werden in Kontext dieser Arbeit folgende Fragen beantwortet:

1. Welche Design Patterns werden berücksichtigt?
2. Was für ein Datensatz eignet sich für solch ein Prozess?
3. Wonach wird exakt klasszifiziert?
4. Welche Merkmale, die aus Quellcode-Dateien extrahierbar sind, eignen sich für Klasszifierung durch ein Maschine Learning Modell?

5. Welche Klassifizierer eignen sich?
6. Wie ist das Endresultat zu beurteilen?

2 Literaturrecherche

2.1 Design Patterns

2.1.1 Design Pattern Katalog

Creational Design Patterns

Structural Design Patterns

Behavioral Design Patterns

2.1.2 Rollenkatalog

2.2 Alternative Ansätze

2.3 Angewendete Ansätze mit Maschine Learning

2.4 Geeiegnetes Datensätze

2.4.1 Verfügbare gelabelte Datetsätze

2.4.2 Argumentieren des Datensatzes mit synthetischen Daten

2.5 Feature Engineering

2.5.1 Transformation von Quellcode in ASTs

2.5.2 Extraktion von Software-Metriken

2.5.3 Umwandlung von Quellcode in textuelle Form

2.6 Betrachte Multiclass-Klasszifizierer

2.6.1 Ein-Modell Architekturen

2.6.2 Mehr-Modell Architekturen

2.7 Metriken für Klasszifizierer

2.7.1 F1

2.7.2 Recall

2.7.3 Precision

2.7.4 ROC

2.8 Angewendete Technologien, Frameworks und Bibliotheken

3 Methodologie

3.1 Verwendeter Datensätze

3.2 Extrahierte Features

3.3 Angewendete Multiclass-Klasszifizierer

3.4 Evaluation des trainierten Models

3.5 Klasszifizierung

3.5.1 Klassifizierung von Rollen in Design Patterns

3.5.2 Klasszifizierung von Design Patterns durch Rollen

3.6 Evaluation der Methodologie

4 Zukünftige Aussichten

A Erstes Kapitel des Anhangs

Wenn Sie keinen Anhang benötigen, dann bitte einfach rausnehmen.

Literaturverzeichnis