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

1.2 Untersuchungsfragen

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

2.6 Betrachte Multiclass-Klasszifizierer

2.7 Metriken für Klasszifizierer

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

Einführung

Einführung in Design Patterns

Bei der Entwicklung von Software-System stößt man auf Probeleme und Situationen, denen man bereits in der gleichen Form oder in einer ähnlichen Variation entgegenkommen ist. Daher tendiert man, einen bereits bekannten Lösungsansatz hier wieder einzusetzen. Die Idee der Wiederverwendbarkeit von bekannten Lösungsanätzen für wiederaufkehrende zu lösende Probleme in der Software-Entwicklung von Software-Systemen stellen den Kern der Entwurfsmunster oder *Design Patterns* dar. Allgemeiner betrachtet bilden Entwurfsmunster eine Art Lösungsblaupause für wiederaufkehrende Probleme dar, die für die jeweilige Situation angepasst werden. *Design Pattern - Elements of Reusable Object-Oriented Software* von Gamma et. al [Gam94] ist wohl das bekannteste Werk, das sich mit dieser Thematik auseinander setzt. Normalerweise werden Design Patterns verwendet, um abstrakte Probleme in der Software-Architektur von objekt-orientierten Programmiersprachen zu lösen, die sich mit der Kreation, Struktur oder Verhalten auseinandersetzen. Obwohl Design Patterns als bekannte und erpropte Blaupause für wiederaufkehrende Probleme in Software-Systemen eingesetzt werden können, bringen Design Patterns Konsequenzen für die zu betrachtende Situation mit sich, die in Erwägung gezogen werden sollten. Diese sind meistens erhöhte Zeit- und Speicherkomplexität als Austausch für eine bessere Flexibilität im Design des Software-Systems, so dass dessen Adaptierbarkeit für neue oder sich änderene Anforderungen sichergestellt werden können.

Die Beschreibung von Design Patterns sind detaliert und beinhalten unteranderem dessen Namen, Absicht, Motivation, Awnwendbarkeit, Struktur, Teilnehmer und deren Zusammenarbeit und andere Informationen. Die Semantik des Design Patterns inkludieren die Abischt, Motivation und Anwendbarkeit, die angeben, was das Entwurfsmuster macht, warum dieses benötigt wird und wo es sinnvoll engegesetzt werden kann. Die Teilnehmer reflektieren deren Natur als Blaupause, da diese Rollen darstellen, die die Klassen im Kontext des Design Patterns darstellen, während die Struktur und die Zusammenarbeit zwischen den Teilnehmern die Interaktionen zwischen diesen beschreibt. Diese Menge an Informationen ist in den Entwurfsmustern un deren Beschreibung enkodiert und wird durch Software-Entwickler in das Software-System während dessen Implementierung engeflochtet. Der Einsatz von Design Patterns ist auf Design-Entscheidung während der Entwicklung des Software-Systems zurückzuführen und des Öfteren wird diese und die jewiligen Hintergrundgedanken nicht weiter dokumentiert. Davon ebenfalls betroffen sind Anpassungen und konkrete Implementierungsdetails des Entwurfsmusters. Schlussendlich tauchen diese Einzelheiten in Quellcode des Software-Systems ab. Das Widerfinden dieser enkodierten Information im Quellcode für die Zwecke der Weiterentwicklung und Wartung ist der Hauptmotivator für die Erkennung von Entwurfsmusterm.

A Erstes Kapitel des Anhangs

Wenn Sie keinen Anhang benötigen, dann bitte einfach rausnehmen.

Literaturverzeichnis

[Gam94] E. Gamma, R. Helm, R. Johnson und J. M. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1 Aufl., 1994.