

Erkennung von Design Patterns durch Machine Learning

1st Mehmet Aslan

Fakultät für Informatik

Hochschule Rosenheim

Rosenheim, Deutschland

mehmet.aslan@stud.th-rosenheim.de

Zusammenfassung—Diese Seminararbeit setzt sich den Fokus, wie mithilfe von Machine Learning das Problem der Erkennung von Design Patterns erläutert. Dabei wird definiert, was unter Design Patterns im Rahmen dieser Seminararbeit zu verstehen und welchen Mehrwert die automatisierte Erkennung von diesen mit sich bringt. Zudem werden Ansätze erläutert, die diese Problematik ohne Einsatz von Machine Learning lösen. Als finaler Abschnitt werden die groben notwendigen Schritte vorgestellt, wie die Erkennung von Design Patterns durch maschinelles Lernen gelöst werden kann. Dabei werden Prozesse vorgestellt, die sich mit dieser Thematik befassen und wie diese den entsprechenden Schritt umgesetzt haben.

Index Terms—Machine Learning, Design Pattern Detection

I. EINLEITUNG

In der Entwicklung von Software-Systemen werden im Öftern auf Probleme gestoßen, die bereits in der gleichen oder in ähnlicher Form bereits gelöst wurden. Für diese gängigen Probleme haben sich im Gegenzug gängige Lösungsblaupausen etabliert, womit sich diese lösen lassen. Dadurch, dass Software-Systeme durch neue oder angepasster Änderungen sich inkrementell weiterentwickeln, ist die strukturelle Verbesserung und Neugestaltung des unterliegenden Quellcodes ohne Veränderung des Programmverhaltens, auch Code-Refactoring genannt, Teil des Entwicklungsprozesses. Bei komplexen Software-Systemen werden auch Design Patterns verwendet. Bei Code-Refactoring muss man den gewissen Systemkontext während des Prozesses im Kopf behalten, um den betrachte Programmfluss im Quellcode zu verstehen. Dabei können Programmstrukturen über mehreren Quellcode-Dateien, Module oder Bibliotheken verteilt sein. Deshalb kann der Erhalt des Systemkontextes im Verständnis schwerfallen, vor allem bei Design Patterns. Um dagegenzuwirken, kann werkzeug-gestützte Hilfestellung mithilfe von Machine Learning den Refactoring-Prozess mit Umgang von Design Patterns einfacher gestalten. Diese Seminararbeit setzt sich die Untersuchung der Erkennung von Design Patterns mit Machine Learning als Fokus. Dabei ist die Arbeit in folgende Segmente untergliedert: gesetzte Einschränkung für die Untersuchung, Ansätze aus nicht Machine Learning Gebieten und die notwendigen Schritte für die Erstellung eines Modells durch Vergleich verschiedener bereits untersuchter Prozesse.

II. EINGRENZUNGEN DER SEMINARARBEIT

Um mit der Erkennung der Design Patterns anzufangen, muss vorerst definiert werden, welche Design Patterns überhaupt zu betrachten sind. Da der Fokus dieser Seminararbeit auf der Analyse von Quellcode liegt, werden hier deren Einsatz in Programmiersprachen betrachtet. Das Werk, was den Begriff der Design Patterns in den Köpfen den Software-Entwicklern am ehesten assoziieren, sind die Konstrukte, die im Werk "Design Patterns: Elements of Reuseable Software" definiert. Diese Design Patterns sind ebenfalls unter dem Begriff "GoF Design Patterns" bekannt. Da der Fokus in diesem Werk den Einsatz von Design Patterns in einem objekt-orientierten Kontext betrachtet werden, wird diese auch so im weiteren Verlauf der Untersuchung übernommen. Obwohl die Erläuterungen der GoF Design Patterns ursprünglich in der Programmiersprache C++ verfasst wurden, wird in dieser Arbeit deren Einsatz ebenfalls in anderen objekt-orientierten Programmiersprachen betrachtet. Dabei werden die Design Pattern in folgende Kategorien unterteilt werden [1, p. 5797]:

Creational	Structural	Behavioral
Abstract factory Builder Factory method Singleton	Adapter Bridge Composite Decorator Facade Flyweight Proxy	Chain of Responsibility Command Interpreter Iterator Mediator Memento Observer State Visitor Strategy Template method

Abbildung 1. Design Pattern Kategorien nach GoF

III. ALTERNATIVE ANSÄTZE

Neben den Lösungsansätzen aus dem Machine Learning Bereich, wurden Verfahren aus anderen Bereichen der Informatik eingesetzt, um die Erkennung von Design Patterns bis zu einem gewissen Grad zu automatisieren. In diesem Abschnitt der Seminararbeit werden einige Verfahren vorgestellt.

A. Strukturanalyse

B. Regelbasierende Analyse

C. Statistische Analyse

IV. ABLAUF DES TRAININGSPROZESSES IM MACHINE LEARNING

A. Data Gathering

B. Data Preprocessing

C. Feature Extraction

D. Model Selection

E. Model Training

F. Model Evaluation

V. ZUKÜNFTIGER AUSBLICK

LITERATUR

- [1] H. Yarahmadi and S. M. H. Hasheminejad, "Design pattern detection approaches: a systematic review of the literature," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5789–5846, Dec 2020. [Online]. Available: <https://doi.org/10.1007/s10462-020-09834-5>