

Lukas Linter - Entwurf und Implementierung eines Assistenzsystems auf Basis statischer Codeanalyse zur Unterstützung von Programmieranfängern

Thema:

Lukas Linter - Entwurf und Implementierung eines Assistenzsystems auf Basis statischer Codeanalyse zur Unterstützung von Programmieranfängern

Art:

[MA](#)

BetreuerIn:

[Alexander Bazo](#)

BearbeiterIn:

Jonas Roberts

ZweitgutachterIn:

[N.N.](#)

Status:

[abgeschlossen](#)

Stichworte:

[Software Engineerings Education](#), [Digital Assistent](#), [Karl Klammer](#), [Assistentssysteme für die Programmierausbildung](#)

angelegt:

2017-07-03

Antrittsvortrag:

2018-02-19

Textlizenz:

[Unbekannt](#)

Codelizenz:

[Unbekannt](#)

Hintergrund

Intelligente Benutzerschnittstellen (Intelligent user interfaces) verfügen über spezielles Nutzer- oder Domänenwissen und unterstützen den Nutzer aktiv bei der Bewältigung seiner Aufgaben. Eine besondere Ausprägung dieser Schnittstellen sind die sogenannten digitalen Assistenten, die in unterschiedlich interaktiver Art und Weise die Wünsche und Bedürfnisse des Nutzers identifizieren und bei Hilfestellung bzw. Führung bei der Interaktion mit der genutzten Anwendung bieten. Eine bekannte Implementierung eines solchen Assistenten ist etwa der aus früheren Versionen der MS Office-Produkte bekannte Karl Klammer. In dieser Arbeit soll das Prinzip eines digitalen Assistenten als zentrale Komponente eines Feedbacksystems umgesetzt werden, das Programmieranfänger während der Implementierung erster Übungsaufgaben unterstützt. Neben konkreten syntaktischen Problemen oder Bugs sind qualitativ schlechter Code sowie funktionale Unvollständigkeit häufige Probleme dieser Zielgruppe. Personalisiertes Feedback, das Aufzeigen und Erklären konkreter Fehler

und Probleme im Programmcode sowie das kollaborative Lösen dieser Probleme sind wesentlicher Teil des konstruktivistischen Lernansatzes, der in den ersten Programmierkursen am Lehrstuhl für Medieninformatik verwendet wird. Während sich ein solches Vorgehen leicht im Rahmen von Präsenzveranstaltungen umsetzen lässt, fehlt beim studentischen Selbststudium die Möglichkeit, zeitnahes und relevantes Feedback zu auftretenden Problemsituationen zu geben. Möglicherweise vorhandene Fehlermeldung der Entwicklungswerkzeuge können von Programmieranfängern nicht interpretiert werden oder bieten keine Ansatzpunkt zum konstruktivistischen Lösen der Probleme.

Zielsetzung der Arbeit

Ziel der Arbeit ist die Entwicklung und Implementierung eines digitalen Assistenten, der die aus Tutorien bekannte Lernsituation so weit möglich auf das studentische Selbststudium überträgt. Auf der Basis statischer Code-Analyse soll Studierenden aufbereitetes Feedback zu aktuell im Code identifizierten Problemen angeboten werden. Der digitale Assistent soll dabei diese Probleme lokalisiert und relevant beschreiben und dem Nutzer eine schrittweise Lösung erlauben. Dabei übernimmt der Assistent eine beratende Rolle und liefert das nötige Hintergrundwissen. Der Studierende löst die identifizierten Probleme selbstständig auf der Basis der vermittelten Informationen.

Die Arbeit untersucht dabei unterschiedliche Fragestellung:

1. Die vorhandene Lernsituation in den Tutorien wird verwendet um empirische Daten zur Auftretenshäufigkeit relevanter Fehler zu sammeln.
2. Auf der Basis vorhandener Literatur werden Designempfehlung für die Gestaltung und Ausprägung Digitaler Assistenten gesammelt und verarbeitet.
3. Mit dem entwickelten System soll in einer abschließenden Evaluation die Akzeptanz sowie die Verwendbarkeit unterschiedlicher Variationen des Systems gemessen werden.

Die Leitfrage der Untersuchung lautet dabei:

„Kann ein benutzer- und kontextspezifisches Assistenzsystem Programmieranfängern bei der Interpretation und Behebung häufiger Coding-Probleme unterstützen?“

Das im Rahmen der Arbeit zu entwickelnde System basiert auf bestehende Werkzeugen zur statischen Analyse von Quellcode. Exemplarisch wird es als Plugin für die Entwicklungsumgebung IntelliJ IDEA sowie die Programmiersprache Java entwickelt. Grundlage für die didaktischen Inhalte sind der OOP-Kurs des Lehrstuhls. Das System soll als Mehrkomponentensystem realisiert werden, bestehend aus dem Plugin in der Entwicklungsumgebung, einem als Wissensbasis fungierenden Hintergrunddienst sowie einer separaten Komponente für den digitalen Assistenten.

Konkrete Aufgaben

- Recherche gängiger Werkzeuge zur statischen Codeanalyse in Java
- Mögliche Anpassung existierender Regelsätze für die Anwendung im OOP-Kurs
- Entwicklung eines Plugins zur Extraktion der Ergebnisse der statischen Codeanalyse aus der Entwicklungsumgebung
- Recherche und Aufarbeitung von Patterns und Antipatterns für die Gestaltung Digitaler Assistenten

- Entwicklung einer Wissensbasis für den Einsatz eines digitalen Assistenten für den OOP-Kurs
- Entwicklung einer graphischen Komponenten für die Darstellung des bzw. Interaktion mit dem Assistenten
- Evaluation des Systems mit einer Stichprobe der potenziellen Nutzergruppe

Erwartete Vorkenntnisse

- Gute Kenntnisse im Bereich des Software Engineerings, insbesondere der Programmierung und Architektur von Anwendungen
- Gute Kenntnisse im Bereich des User Interface-Designs
- Gute Kenntnisse in der Planung, Durchführung und Auswertung von Nutzerstudien
- Grundlegende Kenntnisse in der Verwendung statischer Codeanalyse

Vom Vorteil sind Erfahrungen in der Entwicklung von Plugins für Entwicklungsumgebungen wie z.B. IntelliJ IDEA sowie Kenntnisse über den OOP-Kurs und die begleitenden Tutorien.

Weiterführende Quellen

Nach Absprache mit dem Betreuer.

From:
<https://wiki.mi.ur.de/> - **MI Wiki**

Permanent link:
<https://wiki.mi.ur.de/arbeiten/lukas-linter-digitaler-assistent-programmierausbildung>

Last update: **14.02.2022 08:34**

