



# **Smart Home mit Open Home Automation Bus (OpenHAB)**

Lukas Kiederle  
Dominik Ampletzer  
Daniel Boening  
Fakultät für Informatik

WS 2019/20



## Kurzfassung

Aus Zeit- und Kostengründen beim Entwickeln und Testen von komplexen Systemen werden Tools zur Programmcodevalidierung immer relevanter. Diese Tools ermöglichen das Schreiben von Programmen, welche mathematisch und maschinell geprüft sind. Dadurch ist sichergestellt, dass das beschriebene Programm sich auch wie gewünscht, verhält.

Ziel dieser Arbeit ist es, einen sowohl theoretischen als auch technischen Einblick in die Programmcodevalidierung mit dem Proof Assistent Tool Coq darzustellen. Als Einstieg werden die grundlegende Begriffe geklärt und ein kurzer Überblick über Tools in diesem Fachbereich dargestellt. Dabei wird insbesondere auf Coq eingegangen.

Um ein Verständnis zu bekommen, wie ein Proof Assistent Tool die Qualität von Programmcode sicherstellt, müssen zunächst die Grundlagen dieser Sprache anhand von Beispielen erklärt werden. Anschließend wird näher auf das Zusammenspielen zwischen Programmcode und Proof Assistent eingegangen.

Es gibt bereits einige sehr erfolgreiche Forschungsprojekte, die Coq im Einsatz haben. Diese werden abschließend vorgestellt. Schlussendlich wird ein Fazit inklusive Ausblick in Hinsicht auf die Verwendbarkeit von Proof Assistent Tools gezogen.

Schlagworte:

- Proof Assistant
- Coq
- Programcodevalidierung

## Leseanleitung

Hinweise auf referenzierte Literatur und die daraus entnommenen Zitate, welche in eckigen Klammern angegeben sind, werden im Literaturverzeichnis am Ende der Arbeit aufgeführt. Soll ein Begriff oder eine Formulierung besonders hervorgehoben werden, ist diese *kursiv* geschrieben. Abkürzungen werden bei erstmaligem Auftreten einmal in runden Klammern, anschließend an das Wort ausgeschrieben. Um den Lesefluss nicht zu stören, werden alle darauf folgenden Wiederholungen der Abkürzungen nicht immer explizit ausgeschrieben.

Möglicherweise unbekannte Begriffe und Fachbegriffe werden bei ihrer ersten Nennung **fett** gedruckt. Diese sind im Glossar in alphabetischer Reihenfolge aufgelistet und werden näher erklärt. Einzige Ausnahme hierbei sind Überschriften von Tabellen. Um ein zusammenhängendes Lesen der Arbeit zu erleichtern, werden bei Bedarf Erklärungen bereits im Text gegeben. Dabei wird davon ausgegangen, dass der Leser bereits mit grundlegenden Begriffen der Informatik vertraut ist. Ausgehend vom Wissensstand eines entsprechend vorgebildeten Lesers, werden demzufolge nur fachlich speziellere Begriffe erklärt.

Um Unklarheiten zu vermeiden, werden Fachbegriffe in und zur Beschreibung von Bildern und Prozessen in ihrer originalen Sprache Englisch verwendet und nicht immer übersetzt.

An den Stellen, an denen es der Ausführung des Textes dient, sind kurze Codebeispiele im Text eingebunden. Außerdem werden Abbildungen zur Veranschaulichung verwendet, um komplexe Prozesse einfacher und verständlicher zu machen. Größere Abbildungen befinden sich im Anhang und werden im passenden Textabschnitt referenziert. Damit soll der Lesefluss nicht gestört werden.



# Inhaltsverzeichnis

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Motivation</b>                           | <b>6</b> |
| <b>2</b> | <b>Was ist OpenHAB?</b>                     | <b>6</b> |
| <b>3</b> | <b>OpenHAB aus technischer Sicht</b>        | <b>6</b> |
| 3.1      | Was sind Geräte? . . . . .                  | 6        |
| 3.2      | Was sind Items? . . . . .                   | 6        |
| 3.3      | Geräteerkennung . . . . .                   | 6        |
| <b>4</b> | <b>Exemplarische Verwendung von OpenHAB</b> | <b>6</b> |
| <b>5</b> | <b>Fazit</b>                                | <b>6</b> |
| 5.1      | Stärken . . . . .                           | 6        |
| 5.2      | Schwächen . . . . .                         | 6        |
| <b>6</b> | <b>Infos:</b>                               | <b>6</b> |
| <b>7</b> | <b>Vorlage mit Samples</b>                  | <b>7</b> |
| <b>A</b> | <b>Erster Abschnitt des Anhangs</b>         | <b>9</b> |

# 1 Motivation

- Open Hub im Fach Software Architektur untersuchen und praktischen Nutzen feststellen
- test

# 2 Was ist OpenHAB?

The open Home Automation Bus (openHAB) is an open source, technology agnostic home automation platform which runs as the center of your smart home!

Some of openHAB's strengths are:

Its ability to integrate a multitude of other devices and systems. openHAB includes other home automation systems, (smart) devices and other technologies into a single solution To provide a uniform user interface and a common approach to automation rules across the entire system, regardless of the number of manufacturers and sub-systems involved Giving you the most flexible tool available to make almost any home automation wish come true; if you can think it, odds are that you can implement it with openHAB.

# 3 OpenHAB aus technischer Sicht

## 3.1 Was sind Geräte?

## 3.2 Was sind Items?

## 3.3 Geräteerkennung

# 4 Exemplarische Verwendung von OpenHAB

- Wie ist OpenHAB installiert (OpenHAB cloud oder auf raspi?)
- Welche Geräte haben wir mit OpenHAB verbunden?
- Wie haben wir die Geräte verbunden?
- On the server the configuration is stored somewhere in userdata (/var/lib/openhab2 for apt-get installs).

In an upgrade the userdata folder is preserved when using apt-get.

# 5 Fazit

## 5.1 Stärken

## 5.2 Schwächen

# 6 Infos:

**Ausgangslage** Untersuchen Sie die Architektur und Features von OpenHAB und schreiben Sie ein Beispielanwendung. Mit myOpenHub existiert eine kostenlose Plattform die sie nutzen können.

**Beantworten Sie dabei**

- Aktueller Status des Projekts und
- Integration der Big Player wie Alexa und Google Home
- Welche Tools und Konzepte und APIs gibt es
- Welche Deployment Modi und Betriebsmodi existieren
- Untersuchen Sie auch Aspekte wie Datenintegriertät und Sicherheit

#### Unterlagen Linkes

- <https://www.myopenhab.org/>
- <https://www.openhab.org/>
- <https://jaxenter.de/openhab-2-4-78711>

## 7 Vorlage mit Samples

Codebeispiel:

```

1 public class JavaExample
2 {
3     public static void main(String[] args)
4     {
5         int test = 5;
6         int result = 10;
7
8         result += test;
9     }
10 }
```

Codebeispiel 1: Java Beispiel

Einen Überblick findet man z. B. in [Aue00].

Ein Beispiel wird in Abb. 1 gezeigt. Das verwendete Objekt ist in Abb. 1a dargestellt, das Ergebnis in Abb. 1b.

Eine Formel

$$f(x) = \frac{1}{3}x + 5, \quad x \in \mathbb{R}. \quad (1)$$

Und noch eine:

$$\mathbf{M} = \mathbf{A}\mathbf{x}\pi, \quad \mathbf{A} \in \mathbb{R}^{2 \times 2}, \mathbf{x} \in \mathbb{R}^2. \quad (2)$$

Tabelle 1 gibt einen Überblick über XYZ.

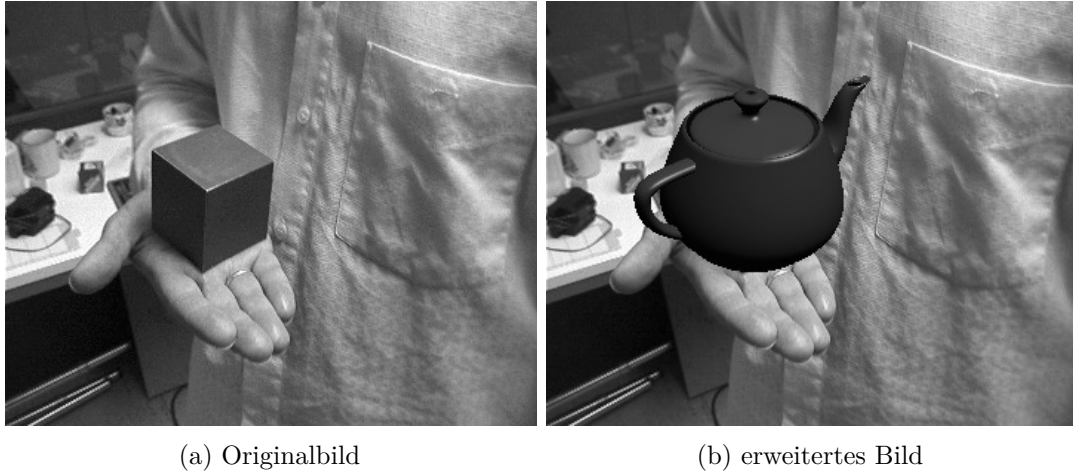


Abbildung 1: Beispiel eines Augmented Reality Systems: es folgt eine Beschreibung (Bilder aus [Sch01])

| Sequence                   | ARTS   | wman  | stcams | ARTVZ       | ARTSUZ  |
|----------------------------|--------|-------|--------|-------------|---------|
| # Frames                   | 190    | 40    | 400    | 270         | 190     |
| # relative movements       | 17955  | 780   | 79800  | 36315       | 17955   |
| # movements after pre-sel. | 14336  | 623   | 37915  | 21788       | 14343   |
| min. angle in seq.         | 0.233° | 5.95° | 0.154° | 0.00000171° | 0.0388° |
| max. angle in seq.         | 81.7°  | 180°  | 47.3°  | 80.3°       | 80.9°   |
| min. angle after pre-sel.  | 12.9°  | 21.1° | 17.3°  | 16.3°       | 12.9°   |
| max. angle after pre-sel.  | 81.7°  | 161°  | 47.3°  | 80.3°       | 80.9°   |

Tabelle 1: Datenselektion für verschiedene Testdatensätze.



## **A Erster Abschnitt des Anhangs**

In diesem Anhang wird . . .

## Literatur

- [Aue00] T. Auer. *Hybrid Tracking for Augmented Reality*. Dissertation, Technische Universität Graz, Graz, Austria, 2000.
- [Sch01] J. Schmidt, I. Scholz und H. Niemann. Placing Arbitrary Objects in a Real Scene Using a Color Cube for Pose Estimation. In B. Radig und S. Florczyk, Hg., *Pattern Recognition, 23rd DAGM Symposium*, Bd. 2191 von *Lecture Notes in Computer Science*, S. 421–428. Springer-Verlag, Berlin, Heidelberg, New York, 2001.