

Usability-Test zur Master-Thesis

Konzeption und prototypische Umsetzung einer Steuerzentrale eines smarten Büros mit dem Fokus einer einfachen Handhabung der formalisierten Interaktionen für Softwareentwickler

Im Rahmen der Master-Thesis zu obigem Thema wird ein Usability-Test durchgeführt, anhand dessen die Nutzbarkeit eines Frameworks ermittelt werden soll. Ziel des Systems ist die einfache Handhabung der formalisierte Interaktionen für Softwareentwickler, um in wenigen Schritten eine lauffähige Steuerzentrale und deren Regelwerk zu implementieren.

Die Aufgabe, die innerhalb des Usability-Tests gestellt wird, ist die Implementierung eines kleinen Anwendungsfalls. Dieser wird in kleinere Teilaufgaben unterteilt und Schritt für Schritt erläutert.

Vorab wird das Framework und deren Kernkomponenten erläutert, sodass der Einstieg zur Aufgabe und Nutzung des Frameworks mit einer gewissen Vorkenntnis stattfindet. Nach der Schilderung und der Klärung aller potentiell auftretenden Fragen wird der Test durchgeführt.

Bedingungen:

Es gibt keine zeitliche Einschränkung, die Probanden unterliegen keinem Zeitdruck. Die Durchführung ist ortsunabhängig und wird an unterschiedlicher Hardware durchgeführt. Lediglich ist sicherzustellen, dass die Kommunikation über MQTT mittels einem vorab zur Verfügunggestellten MQTT-Broker erfolgen kann. Hierfür wird ein Raspberry Pi 3 verwendet, der einen vorkonfigurierten MQTT-Broker bereitstellt. Die ein- und ausgehenden MQTT-Nachrichten werden, sofern keine ansteuerbaren Geräte zur Verfügung stehen, simuliert. Die Performanz der Kommunikation ist während der Testung zu vernachlässigen. Die im Rahmen dieses Testes benötigten Komponenten, darunter der Service-Roboter sind simuliert und geben entsprechend Rückmeldung.

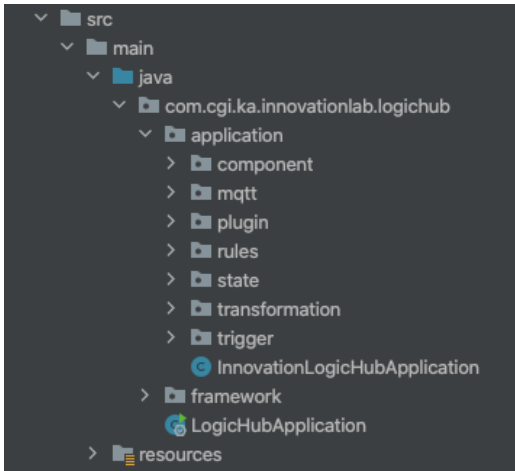
Fragen, die während der Durchführung entstehen, können direkt gestellt werden. Diese werden umgehend beantwortet.

Nach Abschluss des Tests werden die Probanden mittels eines System Usability Scale (SUS) Fragebogen um Rückmeldung gebeten. Abschließend wird noch ein Interview durchgeführt, um weitere Informationen zu erheben. Dabei wird gezielt auf die Erfahrung, sowie die Erweiterung- und ggf. Auf Verbesserungspotentiale eingegangen.



Einleitung zum Usability-Test

Konkrete Kenntnisse über das Spring Framework sind nicht vorauszusetzen. Gestellt ist ein initiales Spring Projekt, welches die Struktur bereits vorgibt.



Die Informationen zur Anbindung des MQTT-Clients an den MQTT-Broker wird begleitend eingerichtet. Unter dem Ordner „resources“ ist in der Konfigurationsdatei die IP, der Nutzernamen und das Passwort des clients anzugeben. Weitere Einstellungen müssen nicht getätigt werden. Anschließend sind die Teilaufgaben selbstständig zu lesen und umzusetzen. Diese werden abstrakt gehalten, um Entscheidungen nicht vorweg zu nehmen.

1.1 Vorgegebene Projektstruktur

Die einzige vorgegebene Klasse ist die der Applikation selbst, von dort aus die Interaktionen und das Framework initialisiert und gestartet wird. Es handelt sich dabei um die „InnovationLogicHubApplication“. Diese stellt ein leeres Konstrukt bereit, das es gilt zu füllen, sodass nach Bearbeitung der Aufgaben dieses auch getestet werden kann. Das leere Konstrukt ist der folgenden Abbildung zu entnehmen:

```
InnovationLogicHubApplication.java
1 package com.cgi.ka.innovationlab.logichub.application;
2
3 import org.springframework.stereotype.Service;
4
5 @Service
6 public class InnovationLogicHubApplication {
7
8     public InnovationLogicHubApplication() {
9         System.err.println("Let's tryout the Innovation Lab KA state implementation");
10         System.err.println("-----");
11     }
12 }
13
```

1.2 Konstrukt zu Implementierung des Frameworks

Innerhalb des Konstruktors „InnovationLogicHubApplication()“ sollen die Teilaufgaben implementiert werden. Mit diesen Voraussetzungen kann die Bearbeitung der einzelnen Aufgaben ohne begleitende Hilfe beginnen.



Aufgabe

Es soll ein Anwendungsfall implementiert werden, bei dem ein Service-Roboter einen Mitarbeiter oder Gast, der an der Tür steht, empfangen und begrüßen soll. Eingangsbedingungen sind, dass eine Kamera zur Authentifizierung für die Anwendung simuliert wird. Die MQTT-Nachricht wird im Rahmen der Aufgabe manuell erzeugt und losgelöst. Auch die Definition des MQTT-Topics ist vorgegeben. Nachdem die simulierte Authentifizierung ausgelöst wurde, soll über die Steuerzentrale eine Regel ausgeführt werden, die folgende Anforderungen erfüllen soll:

- Nach eingehendem Topic soll der Service-Roboter angesteuert und an die Tür geschickt werden. (Die Ansteuerung des Service-Roboters erfolgt ebenso über MQTT. Da in den meisten Fällen kein Roboter verfügbar ist, wird auch diese Kommunikation mittels MQTT simuliert.)
- Ist der Roboter an der Tür, soll er die Begrüßung starten. (Die folgende Interaktion wird im Rahmen des Test nicht durchgeführt. Nach der Nachricht zur simulierten Begrüßung gilt die Aufgabe als erledigt.)

Für die Aufgabe sind folgende Punkte notwendig:

- Die Kenntnis über MQTT-Topics, die für die Kommunikation benötigt werden. (Werden im Rahmen des Test vorgegeben)
- Ein Zustandsraum, der alle benötigten Komponenten abbildet.
- Der Service-Roboter als Komponente, sodass dieser bei Ausführung einer Aufgabe für weitere Aufgaben geblockt werden kann.
- Der Auslöser innerhalb der Regel, welcher durch ein MQTT-Trigger (Enum) abgebildet wird.
- Die Transformation der eingehenden MQTT-Topics zu Zustandsänderungen, auf die Regeln ausgeführt werden sollen.

Besonders zu beachten ist die Definition von Regeln, sowie deren Integration in das System. Das allgemeine Setup kann weniger gewichtet werden.

Die Vorgehensweise wird den Probanden vor der Durchführung nicht aufgezeigt.

Empfehlung für die Vorgehensweise:

1. Transformation mit den Werten (Topic, Key, Value der JSON, optionale White-List, Name des Attributes im Zustandsraum) definieren.
2. Die Transformation dem Framework übergeben (`logicHub.addTransformation(new SimpleTopicTransformation(...))`)
3. Komponente des Service-Roboters erstellen. („extends Component“)
4. Zustandsraum erstellen, der die Werte als auch Getter und Setter der Werte implementiert. („Z“)
5. Regel definieren, die eine bestimmte Bedingung erfüllen müssen, um darauf eine Aktion auszuführen. („extends Rule<Z>“)
 1. Regel Trigger -> Funktion mit Trigger.MQTT belegen
 2. Regel Condition -> Funktion mit Bedienung des Zustandsraumes überprüfen
 3. Regel Process -> Funktion mit den Schritten zu Begrüßung mittels Thread.sleeps

Fragebogen in Form des SUS

Der folgende Fragebogen soll die Erfahrung und das Empfinden, bzw. die Meinung des Probanden während der Nutzung des Frameworks und der soeben abgeschlossenen Aufgabe widerspiegeln. Hierfür wird das System Usability Scale Template¹ verwendet.

Es sind 10 Aussagen, die mit einem Wert von 0-100 beantwortet werden sollen. 0 stellt dabei die Widersprache da und 100 die volle Zustimmung.

Nr.	Aussage	Wert
1	Ich denke, dass ich dieses System gerne öfter nutzen würde.	
2	Ich fand das System unnötig komplex.	
3	Ich fand das System einfach zu bedienen.	
4	Ich denke, dass ich die Unterstützung einer technischen Person benötigen würde, um dieses System nutzen zu können.	
5	Ich fand, dass die verschiedenen Funktionen in diesem System gut integriert waren.	
6	Ich dachte, es gäbe zu viele Inkonsistenzen in diesem System.	
7	Ich könnte mir vorstellen, dass die meisten Leute sehr schnell lernen würden, dieses System zu benutzen.	
8	Ich fand das System sehr umständlich zu bedienen.	
9	Ich fühlte mich sehr sicher mit dem System.	
10	Ich musste viele Dinge lernen, bevor ich mit diesem System loslegen konnte.	

* Die Probanden sind anonym und werden zu keinem Zeitpunkt veröffentlicht. Die Bewertung durch den Fragebogen ist ebenso anonym. Es kann zu keinem Zeitpunkt eine Schlussfolgerung auf die Probanden gezogen werden. Die Durchführung des Usability-Tests, sowie die Auswertung des Fragebogens gilt nur zur Erhebung von Informationen im Rahmen der Nutzbarkeit.

¹ <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>