

PSE – Kick-off

Prof. Bernhard Beckert, Dr. Mattias Ulbrich, Alexander Weigl

Institut für Theoretische Informatik – Anwendungsorientierte formale Verifikation

07.11.2016

TOP

Organisation

Betreuer

Zeitplan

Wöchentliche Treffen

Phasen

Abgabe & Prüfungsleistungen

Werkzeuge

Pflichtenheft

Anforderungen

Ausblick

Betreuer

Alexander Weigl

weigl@kit.edu - Büro 225

Mattias Ulbrich

ulbrich@kit.edu - Büro 229

Webseite

formal.iti.kit.edu/teaching/pse/201617/automation/

- ▶ Termine
- ▶ Dokumente

Zeitplan

Phase	Datum
Erstes Gruppentreffen	heute
Pflichtenheft	14.11. – 04.12.
Entwurf	05.12. – 15.01.
Implementierung	16.01. – 12.02.
z.B. Klausurpause*	13.02. – 19.02.
Qualitätssicherung	20.02. – 12.03.
interne Abnahme	13.03. – 19.03.
Abschlusspräsentation*	20.03. – 26.03.

Variable Anteile

- ▶ **Klausurenpause** (Entscheidung bis vor dem Beginn Implementierung)
- ▶ **Abschlusspräsentation** (in Abhängigkeit mit anderen Gruppen)

Wöchentliche Treffen

Es finden wöchentliche Treffen in der gesamten Runde statt.
Mögliche Termine:

Wochentag	Block					
	1.	2.	3.	4.	5.	6.
Montag	-	-	-	X	X	-
Dienstag	-	-	-	X	X	-
Mittwoch	-	-	X	-	X	-
Donnerstag	-	-	-	X	-	-
Freitag	-	-	X	X	-	-

Phase

- ▶ Phasenverantwortlicher
 - ▶ koordiniert die Arbeit
 - ▶ präsentiert die Ergebnisse im Kolloquium
 - ▶ Zuteilung liegt bei Ihnen
- ▶ Kolloquium
 - ▶ Abschluss einer Phase
 - ▶ Vortrag vom Phasenverantwortlichen
 - ▶ Fragenrunde

Abgabe & Prüfungsleistungen

Abgaben (Artefakte)

- ▶ Abgaben erfolgen stets per E-Mail mit den geforderten Artefakten.
- ▶ Teil der Prüfungsleistung
- ▶ Abgabe n Stunden vor dem Kolloquium

- ▶ Kein: “Es liegt da im Repository”

Kolloquium & Abschlusspräsentation

Teil der Prüfungsleistung (Vortrag + Fragerunde)

Wichtig: Anwesenheit ist Teil der Prüfungsleistung

Bewertung

- ▶ Qualität aller abgegebenen Dokumente
- ▶ Qualität der Kolloquien
- ▶ Qualität der Abschlusspräsentation
- ▶ Erfüllen der minimalen Leistungsmerkmale
- ▶ *Sinnvolle* Erweiterungen über diese Merkmale hinaus
- ▶ Qualität des erstellten Programms
(das schließt u.A. Benutzbarkeit und Robustheit ein)

Werkzeuge

Werkzeugunterstützte SW-Entwicklung

Verwendung von

- ▶ Repository (GIT oder SVN)
- ▶ Issues Tracker
- ▶ Continuous Integration
- ▶ Build-Skripte (Maven, Ant, ...)

Repository angeboten von:

- ▶ ATIS (SVN)
- ▶ Github, Bitbucket, etc.
- ▶ SCC (<https://git.scc.kit.edu>)

Den Betreuer ist Zugriff gewähren.

1. Phase: Pflichtenheft

Pflichtenheft

Artefakt: Pflichtenheft im Umfang: ca. 20 Seiten

Inhalte:

- ▶ Einleitung, Aufgabenstellung
- ▶ Systemmodell und -umgebung
- ▶ Zielbestimmungen (Muss-, Wunsch- und Abgrenzungskriterien)
- ▶ vollständige funktionale Anforderungen; Qualitätsanforderungen
- ▶ GUI-Entwürfe
- ▶ ausführliche Testfallszenarien

Optional: Aufwandsabschätzung

Pflichtenheft

Artefakt: Pflichtenheft im Umfang: ca. 20 Seiten

Inhalte:

- ▶ Einleitung, Aufgabenstellung
- ▶ Systemmodell und -umgebung
- ▶ Zielbestimmungen (Muss-, Wunsch- und Abgrenzungskriterien)
- ▶ vollständige funktionale Anforderungen; Qualitätsanforderungen
- ▶ GUI-Entwürfe
- ▶ ausführliche Testfallszenarien

Optional: Aufwandsabschätzung

Kundenanforderungen

- Ideen -

User Story

- ▶ Der Anwender (Ingenieur) öffnet das Programm und lädt den Source code und eine Spezifikation.
- ▶ Nach dem Auslösen der Verifikation zeigt das Programm das Gegenbeispiel grafisch und tabellarisch an.
- ▶ Der Anwender passt die Spezifikation in einen “Tabellen-basierten Editor” an.
- ▶ Erneutes Aktivieren der Verifikation zeigt die Korrektheit an.
- ▶ Über einen Editor verändert der Anwender den Source Code, durch Einfügen einer neuen Ausgabevariablen.
- ▶ Diese Variable erscheint darauf hin in der Spezifikation.
- ▶ Über das Timing Diagram spezifiziert der Anwender das Sollverhalten.

User Story

- ▶ Der Anwender (**Ingenieur**) öffnet das Programm und **lädt** den **Source code** und eine **Spezifikation**.
- ▶ Nach dem Auslösen der Verifikation zeigt das Programm das Gegenbeispiel grafisch und tabellarisch an.
- ▶ Der Anwender passt die Spezifikation in einen **“Tabellen-basierten Editor”** an.
- ▶ Erneutes Aktivieren der Verifikation zeigt die Korrektheit an.
- ▶ Über einen **Editor** verändert der Anwender den **Source Code**, durch **Einfügen** einer neuen Ausgabevariablen.
- ▶ Diese Variable **erscheint** darauf hin in der Spezifikation.
- ▶ Über das **Timing Diagram** spezifiziert der Anwender das Sollverhalten.

GUI Skizze

ST Verification Studio

```
VAR
  TIMER1: TON;
  TIMER2: TON;
  FF1: RS; (* Flipflop 1 *)
  FF2: RS; (* Flipflop 2 *)
  FF3: RS; (* Flipflop 3 *)
END_VAR

TIMER1 ( IN :=NOT FF1.Q1 , PT :=
T#0.2s );
FF1 (SET := TIMER1.Q);
FF1 (RESET1:= NOT TIMER2.Q);
TIMER2 ( IN :=NOT FF2.Q1 AND
TIMER1.q , PT := T#0.4s );
FF2 (SET := TIMER2.Q);
FF2 (RESET1:= NOT TIMER1.Q);
FF3 (SET:=TIMER1.Q);
FF3 (RESET1:=TIMER2.Q);

If RST_Tisch then
```

#	a	q	o
1	T	F	F
2	T	T	T
3	*	T	F
4	*	*	F
5	*	*	T
6	*	*	F

Timing diagram showing signals a, b, and c over time. The x-axis is labeled with time steps 0 through 7 (top) and 9 through 6 (bottom). Signal a is high from t=2 to t=4, then drops to low. Signal b is high from t=4 to t=5, then drops to low. Signal c is high from t=5 to t=6, then drops to low. The diagram is annotated with points a, b, c, d, e and a blue arrow indicating a transition.

“Handelsüblicher” Editor für *Structured Text*

- ▶ Laden und Speichern von Dateien
- ▶ Syntaxhighlighting
- ▶ Übliche Editoroperationen (Einfügen, Ausschneiden)
- ▶ Anzeige von Fehlermeldungen
- ▶ Folding (Kann)

Spezifikation-Tabelle

Tabellen-Ansicht um generalisierteTM Testtabellen zu bearbeiten

- ▶ Beschreibung der Eingabe- und Ausgabewerte
- ▶ Formelunterstützung
- ▶ Eingabe-Validierung
 - ▶ Variablensignaturen
 - ▶ Zyklen
 - ▶ Syntax
 - ▶ Typkonformität
 - ▶ Rückwärtsreferenzen
- ▶ Annotation und Kommentare

Weitere Informationen zum Konzept nächste Woche.

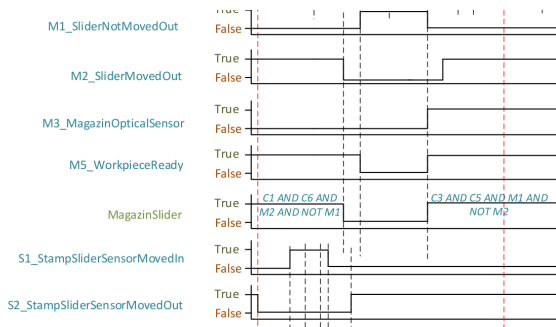
Spezifikation-Tabelle

Tabellen-Ansicht um generalisierteTM Testtabellen zu bearbeiten

- ▶ Beschreibung der Eingabe- und Ausgabewerte
- ▶ Formelunterstützung
- ▶ Eingabe-Validierung
 - ▶ Variablensignaturen
 - ▶ Zyklen
 - ▶ Syntax
 - ▶ Typkonformität
 - ▶ Rückwärtsreferenzen
- ▶ Annotation und Kommentare

Weitere Informationen zum Konzept nächste Woche.

Timing-Diagramm



- ▶ Grafische Darstellung der Spezifikation
- ▶ Unterstützung für Integer, Boolean, Enums
- ▶ Großer Wertebereich?
- ▶ Erlaubt Änderungen durch Drag and Drop

Wie geht es weiter

- ▶ Austausch der Kontaktinformationen
- ▶ Anlegen des Repository
- ▶ Festlegung der Phasenverantwortlichen
- ▶ Einlesen in Structured Text, Java GUI Programmierung
- ▶ **Prüfung der Zulassungsvoraussetzungen**

Bis zum nächsten Wochentreffen!