



TECHNISCHE UNIVERSITÄT BERLIN

Software Engineering for Embedded Systems Group – Prof. Dr. Sabine Glesner

www.pes.tu-berlin.de Secr. TEL 12-4 Ernst-Reuter-Platz 7 10587 Berlin



Softwaretechnik und Programmierparadigmen WiSe 2014/2015

Prof. Dr. Sabine Glesner

Joachim Fellmuth

joachim.fellmuth@tu-berlin.de

Dr. Thomas Göthel

thomas.goethel@tu-berlin.de

Lydia Mattick

lydia.mattick@tu-berlin.de

Tutoren

Achtung: Denkt bitte daran, euch für die Prüfung anzumelden! Falls das über QISPOS für euch nicht möglich ist, meldet das Modul im Prüfungsamt an und gebt den gelben Zettel bei uns im Sekretariat ab.

Hausaufgabenblatt 1

Ausgabe: 21.11.2014

Abgabe: 14.12.2014

Ihr habt den lukrativen Auftrag für eine Software erhalten, welche wissenschaftliche Konferenzen und die Reviews von dafür eingereichten Papers verwalten soll. Der Auftraggeber fordert für die Entwicklung eine visuelle Modellierung in Form von UML-Diagrammen und eine formale Beschreibung unter Verwendung von OCL.

Von eurem Gespräch mit dem Auftraggeber habt ihr folgende funktionale Anforderungen an die Software ermittelt:

Ein Nutzer kann sich unter Angabe seines Namens und Kontaktdaten in der Software registrieren. Nutzer dürfen Vorsitzende von Konferenzen, Autoren und/oder Reviewer von Papers sein.

Die Vorsitzende einer Konferenz kann diese im System registrieren. Dazu werden mehrere Daten benötigt:

- der Name der Konferenz,
- ID bzw. Kurzname,
- der Termin und Ort für die Tagung der Konferenz,
- der Beschreibungstext,

- die erste Deadline - für die Einreichung von Papers,
- die zweite Deadline - für die Reviews der eingereichten Papers und
- die maximale Seitenanzahl für Papers.

Die ID oder Kurzname darf nicht mit einer anderen Konferenz übereinstimmen. Bei den Terminen ist sicherzustellen, dass die zweite Deadline mindestens vier Wochen nach der ersten Deadline ist, und der Konferenztermin selbst wiederum mindestens vier Wochen nach der zweiten Deadline. Spätere Verschiebungen kann die Vorsitzende unter Einhaltung dieser Bedingungen durchführen.

Autoren können ihre Papers in Textform hochladen. Dazu muss zusätzlich der Titel, das Verfassungsjahr und das Abstract eingegeben werden. Die Seitenzahl wird für den Inhaltstext automatisch bestimmt und gespeichert. Der Inhalt des Papers darf nachträglich angepasst werden. Autoren können außerdem alle registrierten Konferenzen einsehen. Ein noch nicht veröffentlichtes Paper darf bei nur einer Konferenz bis zur ersten Deadline eingereicht werden, wenn es die vorgesehene maximale Seitenzahl nicht übersteigt. Ein eingereichtes Paper darf zwischen der ersten und zweiten Deadline und, falls es akzeptiert wurde, nach Abschluss der Konferenz nicht angepasst werden.

Nach Ablauf der ersten Deadline kann die Vorsitzende den Papers Nutzer als Reviewer zuweisen. Ein Reviewer darf nicht Autor desselben Papers sein. Die Auswahl der Reviewer und ihre Anzahl liegt im Ermessen der Vorsitzenden und wird vom System nicht modelliert. Bis zur zweiten Deadline hat ein Reviewer Zeit, ein Review für ein ihm zugewiesenes Paper einzureichen. Es gibt außerdem die Möglichkeit, ein Review abzulehnen, um der Vorsitzenden zu signalisieren, dass ein neuer Reviewer zugewiesen werden sollte. Ein Review beinhaltet einen Text und zusätzlich die Bewertung *accepted*, *borderline* oder *rejected*.

Die Bewertung eines Papers wird aus dessen Reviews berechnet. Jede Reviewbewertung kann in Punkte umgerechnet werden. Die Vorsitzende kann den Punkteschlüssel und die Zusagekriterien anpassen:

- die Punktzahl für ein *accepted* Review, voreingestellt auf 3 Punkte,
- die Punktzahl für ein *borderline* Review, voreingestellt auf 1 Punkt,
- die Punktzahl für ein *rejected* Review, voreingestellt auf -2 Punkte,
- die minimale Gesamtpunktzahl, voreingestellt auf 3 Punkte,
- der minimale Durchschnitt der Punktzahlen, voreingestellt auf 1 Punkte und
- die Option, nur Papers von Autoren zu akzeptieren, welche bereits eine Mindestanzahl von Papers veröffentlicht haben. Diese Einstellung ist voreingestellt auf 0 Papers.

Die Vorsitzende kann nach der zweiten Deadline die Konferenz abschließen. Solange die Konferenz noch nicht abgeschlossen wurde, können sowohl der Punkteschlüssel als auch die Zusagekriterien geändert werden. Die Vorsitzende kann sich alle Papers anzeigen lassen, welche momentan die Zusagekriterien erfüllen: Dadurch kann sie die Anzahl der akzeptierten Papers indirekt beeinflussen, um der Konferenz gerecht zu werden.

Sobald die Vorsitzende eine Konferenz abschließt, werden vom System mehrere Aufgaben erledigt: Alle Papers, welche die Zusagekriterien erfüllen, bleiben mit der Konferenz verknüpft. Die Autoren dieser Papers werden über die Zusage benachrichtigt. Alle übrigen Papers werden von der Konferenz getrennt. Betroffene Autoren erhalten dann die Nachricht, dass ihr Paper bei einer anderen Konferenz eingereicht werden darf. Die von getrennten Konferenzen erhaltenen Reviews werden nicht vom Paper getrennt, beeinflussen aber nicht die Bewertung bei der momentan verknüpften Konferenz. Nach dem Konferenztermin werden alle akzeptierten Papers veröffentlicht.

1. Anwendungsfalldiagramm

Modelliert die Anwendungsfälle des Systems mit einem Anwendungsfalldiagramm. Beachtet dabei folgendes:

- Die Anwendungsfälle dürfen beliebig viele Erweiterungen oder Einbindungen haben.
- Unterscheidet mindestens zwei Akteure.
- Alle im Text beschriebenen Szenarien müssen im Diagramm enthalten sein. Wenn der Inhalt eines Anwendungsfalles nicht eindeutig ist, sollen dessen Szenarien unter dem Diagramm aufgelistet werden.

2. Objektinteraktionsdiagramme (erweiterte Sequenzdiagramme)

Modelliert die folgenden drei Szenarien mithilfe von je einem Objektinteraktionsdiagramm:

- a) **Konferenz registrieren** Hierbei könnt ihr davon ausgehen, dass die Anforderungen an die Termine (Deadlines, Tagung) der Konferenz erfüllt sind.
- b) **Fehlende Reviews** Die Vorsitzende hat die Möglichkeit sich alle Papers ausgeben zu lassen bei denen noch mindestens ein Review offen ist. Hierzu muss lediglich der Name der Konferenz übergeben werden. Beachtet dass dies nur möglich sein soll bevor die zweite Deadline abgelaufen ist.

Beachtet dabei folgendes:

- Aktivitätszonen sind **nicht** gefordert.
- **Hinweis:** Es ist ggf. hilfreich, das Systemklassendiagramm parallel zu den Objektinteraktionsdiagrammen anzufertigen.

3. Systemklassendiagramm

Modelliert die statische Struktur des Systems mit einem Systemklassendiagramm. Beachtet dabei folgendes:

- Wenn die Funktionalität einer Assoziation oder Klasse nicht eindeutig ersichtlich ist, fügt eine kurze Beschreibung unter dem Diagramm hinzu.
- Wie im Tutorium besprochen sollen die Klassen mit den Bezeichnungen <<Boundary>>, <<Control>> oder <<Entity>> annotiert sein.
- Teilt die Funktionen sinnvoll auf zwei oder mehr <<Control>>-Klassen auf.

4. Invarianten

Präzisiert eure Modellierung formal mithilfe von Invarianten in OCL. Die folgenden Invarianten sollen ausgedrückt werden:

- a) Die bei einer Konferenz eingereichten Papers sind beschränkt durch die eingestellte Seitenzahl.
- b) Nachdem die Vorsitzende die Konferenz abgeschlossen hat sind nur noch jene Papers mit einer Konferenz verknüpft, welche alle Zusagekriterien dieser Konferenz erfüllen.
- c) Reviewer dürfen nur Papers bewerten, welche ihnen zugewiesen wurden. Das Einreichen von Reviews ist nur zwischen der ersten und zweiten Deadline erlaubt.
- d) Kein Reviewer darf sein eigenes Paper reviewen.

Beachtet dabei folgendes:

- Wenn erforderlich, dürft ihr das Datumsobjekt `TODAY` zum Vergleich mit dem heutigen Tag benutzen.
- Fügt jeder Invariante eine kurze Beschreibung hinzu.

5. Contracts in OCL

Definiert folgende Funktionen mithilfe von Vor-und Nachbedingungen (Contracts) in OCL:

- a) Ein hochgeladenes Paper soll vom Autor bei einer Konferenz eingereicht werden.
- b) Ein hochgeladenes Paper soll vom Autor noch einmal angepasst werden. Hierfür kann die im System global verfügbare Funktion `getNumberOfPages(input: String):Integer` verwendet werden.
- c) Die Vorsitzende weist einem Paper einen Reviewer zu. Ein Paper und ein Nutzer wird übergeben. Es kann davon ausgegangen werden dass das Paper und der Nutzer im System vorhanden sind.

- d) Ein Reviewer reicht ein Review für ein Paper ein.
- e) Die Vorsitzende einer Konferenz beendet diese.

Beachtet dabei folgendes:

- Alle Anforderungen aus dem Text bzgl. dieser Operationen müssen berücksichtigt werden.

6. Hoare Kalkül

Gegeben ist ein Programm. Beweist mithilfe der Ableitungsregeln des Hoare Kalküls, dass das Programm nach Ausführung die Nachbedingung erfüllt.

```

{y ≥ 0 ∧ m = y}
  z := 1;
  while y > 0 do
    z := z * x;
    y := y - 1;
  {z = xm}
```

Beachtet dabei folgendes:

- Fügt jedem Beweisschritt eine Erklärung hinzu bzw. nennt die angewendete Regel.

Generelle Hinweise und Abgabeformalitäten

- Wir vergleichen eure Abgabe **nicht** mit einer Musterlösung, da es meist mehrere gültige Lösungsmöglichkeiten gibt. Achtet darauf, dass das Gesamtmodell über die Aufgaben hinweg konsistent ist und dem Beschreibungstext entspricht.
- Fügt eurer Abgabe eine Titelseite mit euren Namen, dem Namen eures Tutors und eurem (offiziellen) Tutoriumstermin hinzu.
- Wiederholer von „MPGI 3 - Softwaretechnik“, welche ihre Hausaufgabennote behalten wollen, schreiben ihren Namen / ihre Matrikelnummer **nicht** auf die Abgabe.
- Diese Hausaufgabe ist eine Prüfungsleistung! Wenn ein (gruppenübergreifendes) Plagiat entdeckt wird, resultiert dies in einem Fehlversuch.
- Die Abgabe erfolgt pünktlich über Isis, als PDF bzw. als Astah-Datei und PDF.
- Falls Operatoren aus OCL/UML verwendet werden, die in der Vorlesung bzw. im Tutorium nicht besprochen wurden, müssen diese erläutert und eine Quelle dafür angegeben werden.