

## Praktikum des Übersetzerbaus WS 2003/04: Generierung von Benutzungsoberflächen

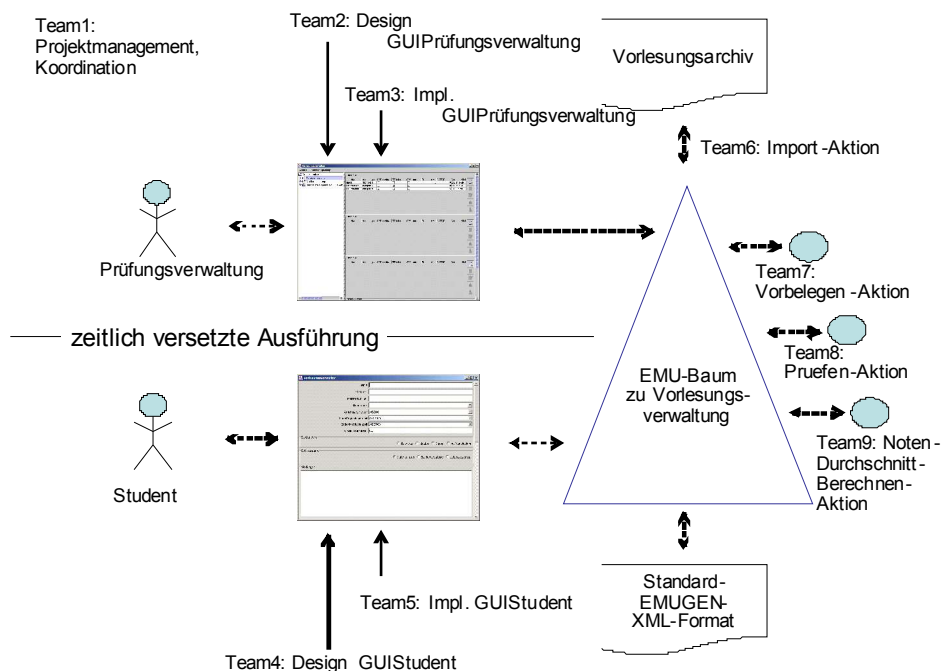
### Aufgabenblatt 8

14.1.2004

### Abschlussaufgabe: Projekt „Vorlesungsverwaltung“

Ausgehend von einer schon entsprechend vorbereiteten Ausgangsbasis (emugen-Eingabe + Rahmen-Javaklassen mit ant-build-Script) sollen von den einzelnen Teams Teilaufgaben zur Erstellung eines Programms zur Vorlesungsverwaltung durchgeführt werden.

### Überblick



Die Anwendung soll aus zwei GUIs/Formularen bestehen.

- „GUI Prüfungsverwaltung“ dient einem Prüfungsverwalter dazu, Daten zu den Vorlesungen zu pflegen bzw. aus einem bestehenden Archiv zu importieren.
- „GUI Student“ dient einem Studenten dazu, abhängig von seinen speziellen Eigenschaften (z.B. Studiengang und Schwerpunkt) und dem aktuellen Vorlesungsangebot (beides wird im EMU-Baum zu Vorlesungsverwaltung verwaltet) eine Vorbelegung seiner Semesterauswahl durchzuführen und anschließend seine/ihre Wahlmöglichkeiten durchzuführen. Durch die Ausführung einer Prüfen-Aktion soll dem Student angezeigt werden, ob seine aktuelle Auswahl ausreichend ist (z.B. ob die ECTS-Punkte genügen). Der Student kann auch durch die Ausführung der Aktion NotendurchschnittBerechnen seinen bisherigen Notendurchschnitt ausrechnen lassen.

Beide GUIs sollen hier zeitversetzt in verschiedenen JVMs auf inhaltlich teilweise identischen EMU-Bäumen arbeiten. Dies soll hier dadurch erreicht werden, dass von beide GUIs dieselbe Standard-EMUGEN-XML-Datei für den Typ Vorlesungsverwaltung geladen wird. Bzw., wenn eine XML-Datei von der Prüfungsverwaltung „freigegeben wurde“ kann es pro Student eine Kopie davon geben, weil der Student die von ihm/ihr gewählten Vorlesungen auch abspeichern muss.

In der vorgegebenen Ausgangsbasis können alle Teilaufgaben als eigene Klassen bzw. Methoden wieder gefunden werden, so dass die entsprechende Aufgabenteilung bzw. die Integration der Lösungen relativ einfach sein sollte. Dennoch ist hierzu erfahrungsgemäß einiges zu tun, so dass dies als eigenständige Aufgabe (Team 1) formuliert wird.

## **Aufgabenverteilung**

Im Weiteren erfolgt stichpunktartig die Aufgabenbeschreibung der einzelnen Teams. Jedes Team ist für die Erledigung seiner Teilaufgabe verantwortlich aber auch für die dazu notwendige Kommunikation mit den anderen Teams.

### **Team 1: Projektmanagement und Koordination**

- Zentrale Code-Verwaltung (siehe Ausgangsbasis)
- Wöchentliche Überprüfung der Fortschritte der Teilteams,
- Nachfragen bei Teilteams nach Problemen,
- Überprüfung der team-übergreifenden Kommunikation
- (nur in Ausnahmefälle sollen Abstimmungsprobleme an Fr. Höllerer bzw. M. Petter gehen)
- Leitung der Abschlusspräsentation

### **Team 2: Design „GUI Prüfungsverwaltung“**

- „Wie muss „GUI Prüfungsverwaltung“ der Ausgangsbasis geändert werden?“
- Anforderungsanalyse
- Design Layout „GUI Prüfungsverwaltung“
- Design Ablauf „GUI Prüfungsverwaltung“
- Welche Felder müssen en/disabled werden?
- Anforderungen für die Vorbelegung, Prüfung und NotendurchschnittBerechnen
- Abstimmung mit Team 3, 7, 8
- Vorstellung „GUI Prüfungsverwaltung“ bei Abschlusspräsentation

### **Team 3: Implementierung „GUI Prüfungsverwaltung“**

Ausgangspunkte:

- vorlesungsverwalter.layout.VorlesungsverwalterLayoutDelegator
- form-Blöcke der emugen-Eingabe der Ausgangsbasis
- vorlesungsverwalter.GUIPruefungsverwaltung

### **Team 4: Design „GUI Student“**

- „Wie muss GUIStudent der Ausgangsbasis geändert werden?“
- Anforderungsanalyse GUI-Student
- Design Layout „GUI Student“
- Design Ablauf „GUI Student“

- Welche Felder müssen en/disabled werden?
- Anforderungen für die Vorbelegung und Prüfung
- Abstimmung mit Team 3, 7, 8
- Vorstellung „GUI Student“ bei Abschlusspräsentation

### **Team 5: Implementierung „GUI Student“**

Ausgangspunkte:

- vorlesungsverwalter.layout.StudiumsplanungLayoutDelegator
- form-Blöcke der emugen-Eingabe der Ausgangsbasis
- vorlesungsverwalter.GUIStudent

### **Team 6: Implementierung „Aktion Importieren Vorlesungsdaten“**

(Info: Um ein ausführbares System zu haben, müssen die Vorlesungsdaten in den emu-Baum für die Vorlesungsverwaltung importiert werden können).

Ausgangspunkt:

- vorlesungsverwalter.util.VorlesungsverwalterActionDelegatorActionDelegator.importieren(..)

### **Team 7: Implementierung „Aktion Vorbelegung Vorlesungsauswahl“**

Zu Beachten: Schwerpunkt-Auswahl, Vorlesungs-Markierung mit Schwerpunkt,

Ausgangspunkt:

- vorlesungsverwalter.util.StudiumsplanungActionDelegator.vorbelegen(..)

### **Team 8: Implementierung „Aktion Prüfen Vorlesungsauswahl“**

Ausgangspunkt:

- vorlesungsverwalter.util.StudiumsplanungActionDelegator.pruefen(..)

### **Team 9: Implementierung „Aktion Notendurchschnitt Berechnen“**

Ausgangspunkt:

- vorlesungsverwalter.util.StudiumsplanungActionDelegator.notendurchschnittBerechnen(...)

### **Hinweise:**

- Die Vorgabe zur Abschlussaufgabe ist hier: [abschlussaufgabe.jar](#)
- Für Teams, die sich mit der Prüfungsverwaltung befassen (Team 1, Team2, Team3 und Team6) ist hier die Vorgabe und die externen Vorlesungsdaten: [abschlussaufgabemitdaten.jar](#)

### **Termine:**

- **Treffen jeden Mittwoch**
- **3.2.2004 Fertigstellung der Lösungen**
- **11.2.2004 Abschlussvorführung**