University of Zurich[UZH]

s.e.a.l. software evolution & architecture lab

Software Praktikum 2012, Übung 2

Architecture Design and Prototyping

1 Informationen

1.1 Kontext

In Übung 1 lernten Sie das zu implementierende Spiel *Asara* kennen, indem Sie mittels Use Cases die Spielabläufe und -regeln erfassten und dem Domainmodel das Spiel und dessen Entitäten modellierten. Sie sollten nun in der Lage sein, dass Spiel vollständig zu verstehen und alle Anforderungen für das spätere System zu nennen. Übung 2 geht einen Schritt weiter in Richtung Implementierung. Ziel ist es mit Hilfe der von Ihnen erstellten Artefakte, die Architektur und das Implementierungsdesign Ihres Spiels zu entwerfen.

1.2 Bestandteile der Abgabe

- Architekturmodell als UML-Klassendiagramm inklusive Beschreibung.
- Implementierungsdesign als UML-Klassendiagramm.
- Lauffähige Prototypen, welche die wesentlichen Teile Ihres Spiel abdecken (Server, Client, RMI, etc.). Im Falle des GUI sind auch Screenshots oder "Zeichnungen" möglich. Diese sollten allerdings schon relativ fortgeschritten sein. Falls Sie sich entschliessen, spezielle Technologien z.B. Animation, Drag'n'Drop zu verwenden, müssen Sie jedoch lauffähige Prototypen haben. Lauffähige Prototypen bzw. deren Source Code müssen ins Repository eingecheckt werden. Im PDF-Dokument, welches Ihr Architekturmodell und Implementierungsdesign enthält, soll ein kurzer Abschnitt über Ihre Prototypen enthalten sein. Dieses Kapitel enthält eine Liste, der von Ihnen erstellten Prototypen und eine kleine Beschreibung jedes Prototypen. Falls Sie an Stelle von lauffähigen Prototypen Screenshots und/oder Zeichnungen erstellt haben, fügen Sie diese Artefakte in dieses Kapitel mit Beschreibung ein.

1.3 Formales

Ihre Abgabe hat in Form einer **einzigen** PDF-Datei per Email an Ihren Betreuer und als Kopie an **hert@ifi.uzh.ch** zu erfolgen. Die Abgabe muss ein Deckblatt mit den Namen und Matrikelnummern aller Gruppenmitglieder beinhalten. Achten Sie darauf, dass das Dokument leicht gelesen und ausgedruckt werden kann; Im Speziellen soll das Dokument nicht Hoch- und Querformat mischen und Grafiken dürfen nicht mehr als eine A4-Seite umfassen. Die Abgabe hat **spätestens bis um Mitternacht vor den Reviewtermin** zu erfolgen. Diese Vorgaben **müssen** eingehalten werden!

1.4 Gruppen

Die Übungen sind ausschliesslich in den gebildeten Gruppen zu lösen. Im abgegebenen Dokument muss ersichtlich sein, wer welchen Beitrag geleistet hat. Ausserdem muss jedes Gruppenmitglied die Abgabe insgesamt verstanden haben und in der Lage sein darüber Auskunft zu geben.

2 Aufgabenstellung

2.1 Architekturentwurf

Der Architekturentwurf resultiert in der Software-Architektur. Diese beschreibt die Organisationstruktur Ihres Systems und stellt somit einen ersten Schritt hin zur Lösung dar. Verwenden Sie einen objektorientierten Architekturstil, und benutzen Sie ein UML-Klassenmodell für die Darstellung. Ein solches Modell beinhaltet ein graphisches Klassendiagramm, welches die vorhandenen Klassen und deren Beziehungen zeigt. Des weiteren sollte eine Beschreibung bzw. Definition zu den einzelnen Klassen vorhanden sein. Diese kann in strukturierter natürlicher Sprache erfolgen. Im Gegensatz zum Domänenmodel enthält ein Architekturmodell auch nicht domänen-spezifische Aspekte wie z.B. Kommunikation (Client-Klassen, Server-Klassen, Proxy-Klassen, GUI-Klassen, Datencontainer-Klassen, etc.). Gehen Sie dabei von Ihrem vorhandenen Domänenmodell aus. Verwenden Sie dabei die Möglichkeiten, die Ihnen die Objekt-Orientierung bietet: Modularisierung, (Daten-)Abstraktion, Information Hiding, Vererbung, Polymorphismus, etc.

Am Schluss sollte ein – oder mehrere – UML-Klassendiagramm(e) resultieren. Es kann sinnvoll sein, jeweils Klassendiagramme für einzelne Komponenten Ihres System zu fertigen. Falls Sie sich für mehrere Klassendiagramme entscheiden, dokumentieren Sie ebenfalls die Komponentenstruktur Ihres Systems mittels eines UML-Komponentendiagramm. Es muss möglich sein, den Code und die Implementierung als bruchfreie Umsetzung der Architektur zu realisieren.

2.2 Implementierungsdesign

Während ein Architekturmodell die Organisationstruktur des Systems als Ganzes betrachtet, stellt das Implementierungsdesign ein Detailentwurf des Systems dar: Detaillierte Beschreibung der Schnittstellen und Charakteristika der einzelnen Komponenten und Entitäten bis hin zu algorithmischen und datenstrukturtechnischen Details. Dies dient als direkter Ausgangspunkt für die Implementierung (vom Prototypen). Dabei können Sie Ihr Klassenmodell verfeineren oder bereits einen groben (evt. dokumentierten) Coderahmen erstellen. Der Code selbst ist eine 1:1 Umsetzung des Implementierungsdesign. Für gewisse Teile z. B. Algorithmen, können Sie bereits im Implementierungsdesign auf Prototypen zurückgreifen.

2.3 Prototypen

Da für die meisten Gruppenmitglieder die eingesetzten Technologien wie RMI oder komplexe GUIs neu sind, ist es sinnvoll sich mittels Prototyping ein besseres Verständnis zu erarbeiten. Dazu sollen sich die einzelnen Gruppenmitglieder mit verschiedenen Technologien auseinandersetzen und einen lauffähigen Prototypen erstellen. Wichtig dabei ist, dass auch die anderen Mitglieder, welche nicht am Bau des jeweiligen Prototypen beteiligt waren, ein gutes Verständnis für die entsprechenden Technologien entwickeln. Nur so ist es möglich die Integration der verschiedenen Bestandteile des zu entwickelnden Systems sicherzustellen. Benutzen Sie bereits hier JUnits Test, um die Funktionalität Ihrer Prototypen zu testen. Für die finale Implementierung **müssen zwingend** Tests vorhanden sein.