

Softwaregrundprojekt

Servicegruppe Informatik | Institut für Softwaretechnik und Programmiersprachen | WiSe 2020/21
23.12.2020 Florian Ege

Meilenstein 2: Anwendungsfälle und Abläufe

Abgabetermin: Mittwoch, 13.01.2021, 18:00 Uhr

Aufgabe 1: Anwendungsfälle

Um einen Überblick über die Interaktionen von Akteuren mit einem System zu bekommen, beschreibt man Anwendungsfälle. Diese werden graphisch als UML2-Anwendungsfalldiagramme dargestellt. Anwendungsfälle fassen meist einige der Funktionalen Anforderungen zusammen. Dabei sollte die Granularität geeignet gewählt werden.

- Fertigt ein **UML2-Anwendungsfalldiagramm** für *Marvelous Mashup* an. Geht dabei von den in Meilenstein 1 identifizierten Akteuren mit ihren Rollen, und den dadurch motivierten Anforderungen aus. Der Übersichtlichkeit halber könnt ihr klar getrennte Anwendungsfälle (bspw. Funktionalität des Editors oder der KI) in einzelne Diagramme auslagern.
- Achtet auf die **Gruppierungsbeziehungen** zwischen den Anwendungsfällen: wo sinnvoll sollten include/extends-Relationen spezifiziert werden. Begründet in textueller Form, warum ihr die Gruppierungsbeziehungen zwischen Anwendungsfällen so gewählt habt.
- Listet für jeden Anwendungsfall auf, welche **Anforderungen** er abdeckt. Verwendet dazu z.B. die IDs, die ihr in Meilenstein 1 vergeben habt.

Aufgabe 2: Abläufe im System

Ausgehend von den eben beschriebenen Anwendungsfällen sollen nun die zugehörigen Abläufe und Prozesse, die man mit dem System durchführen kann, erfasst und dokumentiert werden. Dabei gibt es natürlich mindestens einen Ablauf pro Anwendungsfall, es kann aber auch mehrere geben. Um zeitliche Abläufe in UML2 zu modellieren, stehen zwei hauptsächliche Diagrammtypen zur Verfügung:

Sequenzdiagramme sind geeignet, um Abläufe zu modellieren, in denen mehr als ein Akteur beteiligt ist. Für jeden Teilnehmer gibt es eine Lebenslinie, auf der seine Aktivitäten dargestellt werden. Kommunikation über synchrone Aufrufe oder asynchrone Nachrichten wird durch verschiedenartige Pfeile ausgedrückt. Es gibt Konstrukte für Alternativen und Wiederholungen.

Zustandsdiagramme verwendet man, um stark verzweigte und iterative Abläufe darzustellen. Sie beschreiben einen Ablauf aus Sicht eines einzelnen Akteurs (eben die Zustände, die dieser durchläuft). Möchte man Interaktionen von mehreren Akteuren ausdrücken, sollte man ein eigenständiges Zustandsdiagramm für jeden anlegen, und die Verknüpfung über Nachrichten-Events (mit der Syntax *“Event[Condition]/Action”*) an Zustandsübergangskanten realisieren.

- Fertigt **Sequenzdiagramme** oder **Zustandsdiagramme** für die Anwendungsfälle von *Marvelous Mashup* an. Wählt dabei den Diagrammtyp entsprechend der Art des Ablaufs und des Scopes (der beteiligten Akteure und Komponenten). Auch eine hierarchische Strukturierung in Unterdiagramme ist möglich für bessere Übersichtlichkeit.

Hinweise

- Macht euch vor der Erstellung der einzelnen Diagramme mit der Notation, also der graphischen **Syntax**, und der zugehörigen **Semantik** für die einzelne Diagrammtypen vertraut. Schaut euch dazu besonders die bisher gemachten Übungen nochmals an.
- Wenn Diagramme zu komplex werden, kann man sie auch in einzelne **Teildiagramme** zerlegen, die bestimmte Phasen oder Aspekte eines umfangreichen Ablaufs darstellen. Auch eine hierarchische Gliederung ist oft sinnvoll: ein high-level Diagramm und mehrere **Unterdiagramme**, die Teile davon detaillierter darstellen.
- Versucht, eine sinnvolle **Abstraktionsstufe bzw. Granularität** für die Anwendungsfälle oder Aktivitäten in Abläufen zu finden. "Das Spiel spielen" ist viel zu grob, umgekehrt soll aber auch nicht jeder einzelne Mausklick im Diagramm enthalten sein.
- UML2-Diagramme können viel, aber nicht alles ausdrücken, was zum tiefgehenden Verständnis eines Systems nötig ist. Informationen, die sich gar nicht, oder nur umständlich im Diagramm graphisch darstellen lassen, sollte man darum als **textuelle Annotationen** beifügen, und nicht einfach weglassen.
- Oft denkt man nur an "normale" Situationen. Besonders wichtig ist aber, das auch spezifiziert wird, was bei **Abweichungen vom Regelfall** passieren soll. Achtet darum beim Modellieren auch auf Alternativen, Ausnahmen oder Fehlerfälle.
- Wie immer, schaut euch vor Bearbeitung dieser Aufgaben noch einmal die Korrekturen zu den gleichartigen Einzelübungsaufgaben an.