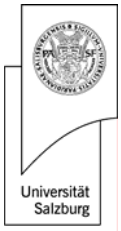
UML fürs Pflichtenheft



Sebastian Fischmeister
Department of Computer Science
University of Salzburg, Austria

Sebastian.Fischmeister@cs.uni-salzburg.at

Overview

- Use-Case Diagramm
- State-Machine Diagramm

Einführung Use Cases

- Eignen sich um die funktionalen Aspekte eines Systems zu erfassen.
- Beschreiben die Interaktion zwischem dem Benutzer und dem System und **nicht** wie das System intern funktioniert.
- Use Cases werden oft über ein Szenario entwickelt:

> *Der Benutzer möchte gerne eine Notiz schreiben. Er logt sich in das System ein und erstellt die Notiz mit einer Kategorie. Automatisch erscheint die Notiz auch bei allen anderen Personen, die diese Kategorie subskribiert haben.*

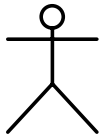
- Ein Szenario ist dabei eine Abfolge von Einzelschritten, die die Interaktion zwischem den Benutzer und dem System beschreibt.

Einführung Use Cases

- Szenarien können unterschiedliche Verläufe haben:
  - Das Einfügen der Notiz wird vom Benutzer abgebrochen.
  - Der Benutzer markiert die Notiz als „Privat“.
  - Ein anderer Benutzer hat einen Filter gesetzt und sieht diese neue Notiz nicht.
- Ein Use Case ist ein Set von Szenarien, die durch ein gemeinsamens Ziel des Benutzers vereint werden (siehe auch das Use Case Formular).

Actor

- Benutzer oder Organisationen werden als „Actor“ bezeichnet.
- Das zugehörige UML Symbol sieht so aus:



- Beispiele eines Actors:
  - Kunden
  - Verkaufsmanager
  - Bank
  - ...
- Ein Actor kann in vielen Use Cases beteiligt sein.
- Der „primary Actor“ ist die hauptsächlich vom Ziel des Szenarios betroffene Person → muss nicht der Initiator sein!

Erfassen eines Use Cases

- Es gibt keinen definierten Standard, aber eine Reihe von Formularen.
- Normalerweise wird das **grundlegende erfolgreiche Szenario** gewählt und die andere Szenarien über Erweiterungen abgebildet.
- Beispiel:

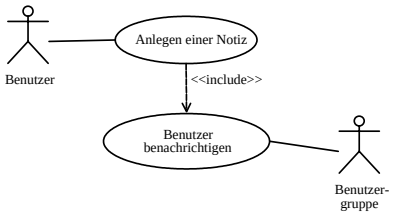
Notiz Einfügen

GES:
1. Der Benutzer logt sich ein
2. Der Benutzer fügt eine neue Notiz ein
3. Die Notiz wird bei allen anderen Benutzern mit derselben Kategorie sichtbar

Erweiterungen:
1a Der Login schlägt fehl
- .1 Das System zeigt den Fehler an
- .2 Das System meldet den Fehler an den Admin
- .3 Der Benutzer logt sich erneut ein

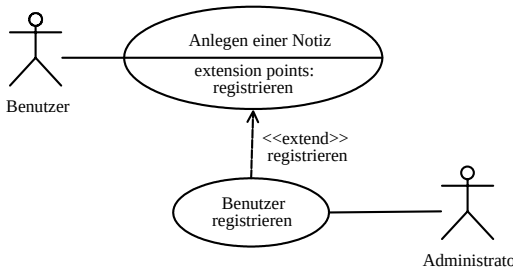
Inkludieren eines anderen Use Cases

- Ein komplizierter Schritt in einem Use Case kann ein eigener Use Case sein:
  - zb. Wenn der Benutzer noch nicht existiert, das Registrieren eines neuen Benutzers
- Kann in textuellen Dokumenten gut mit Hyperlinks abgebildet werden.
- Graphisch wird es durch <<include>> abgebildet.



Erweiterungen für Use Cases

- Erweiterungen sind „Extension Points“
- Sie geben Änderungen zum GES dar:
- zB:



Use Cases: Diverses

- Eine „pre-condition“ für eines Use Cases beschreibt, in welchem Zustand das System sein soll, bevor der Use Case startet.
- Eine „guarantee“ beschreibt, welchen Zustand das System haben soll, nachdem der Use Case beendet wurde.
- Ein „trigger“ beschreibt, mit welchem Ereignis der Use Case gestartet wird.

- „system use case“ und „business use case“.
  - System use case: beschreibt Vorgänge innerhalb der Software
  - Business use case: beschreibt Vorgänge als Reaktion auf Kunden oder Ereignisse
- Cockburn beschreibt Ebenen für use cases:
  - Sea-level, fish-level & kite-level

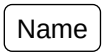
State Machine Diagrams

Einführung

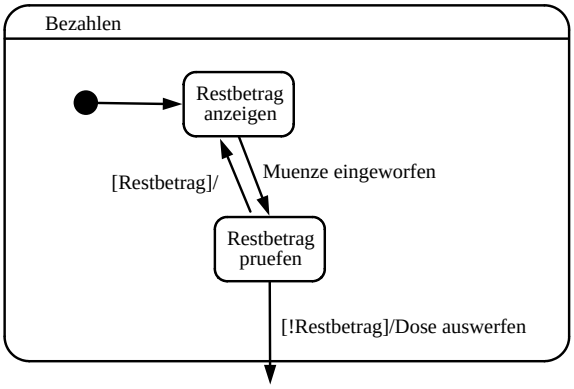
- State Machines beschreiben das Verhalten des Systems.
- Die Wurzeln liegen irgendwo in den 1960-er Jahren und verschiedene Beschreibungsarten wurden entwickelt.

- Beispiele von State Machines sind:
  - Das Verhalten einer einzelnen Klasse
  - Das Verhalten eines ganzen Systems
  - Die Abfolge und Effekte einer Interaktion

UML Elemente

- Initialer Status 
- Übergang 
  - Jeder Übergang kann ein Label haben: *trigger-signature [guard]/activity*
    - Trigger-signature: das auslösende Ereignis
    - Guard: muss zu Wahr evaluieren, damit der Übergang genommen wird
    - Activity: Das Verhalten, das während des Übergangs getätigt wird
- Finaler Status 
- Normaler Status 

Beispiel



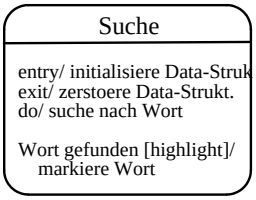
Aktivitäten

- States können auf Events reagieren, ohne einen expliziten Übergang zu benutzen → interne Aktivitäten
- Eine interne Aktivität ist equivalient zu einer Selbst-Transistion

- Darüber hinaus gibt es noch eine Reihe von speziellen Aktivitäten:
  - Entry/... Wird ausgeführt, wenn der Status betreten wird
  - Exit/... Wird durchgeführt, wenn der Status verlassen wird

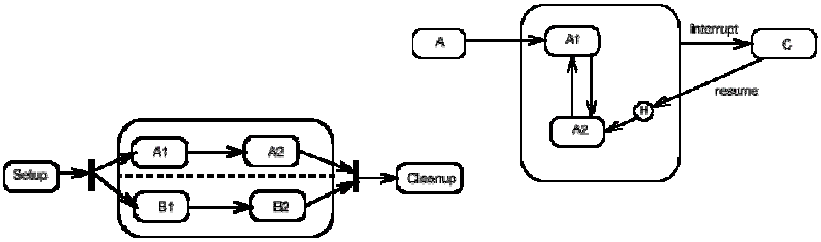
- Interne Aktivitäten lösen keine speziellen Aktivitäten aus !

- States sind normalerweise aktivitätslos. Mit „do/...“ können Aktivitäten spezifiziert werden

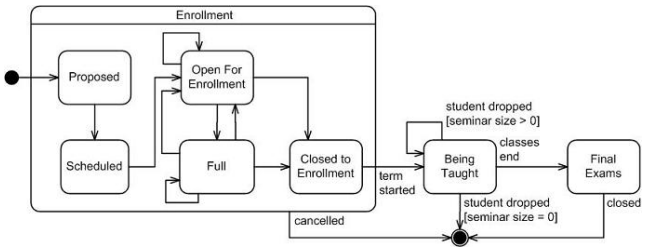


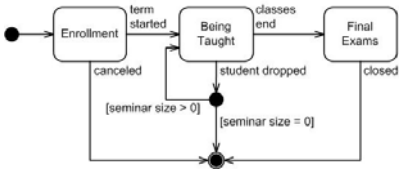
Concurrent States & History States

- State Machines können in UML auch in mehreren Zuständen gleichzeitig sein.
- Der History State zeigt an, welcher der Startzustand im Diagram ist. Der History State wird behalten, wenn der State verlassen wird und wird benutzt, wenn der State das nächste Mal wieder besucht wird.



Beispiele





Referenzen

- http://etna.int-evry.fr/COURS/UML/notation/notation9a.html
- http://www.agilemodeling.com/artifacts/stateMachineDiagram.htm