

Ein Begriffslexikon anlegen und entwickeln Von der Aufgabe ausgehen, nicht von ihrer Lösung Daten suchen, nicht Programmabl" aufe beschreiben Abstraktionsebene nicht in einer Darstellung wechseln Die Spezifikation nach Aspekten organisieren Ein Mengengerüst bilden Den Kunden (Benutzer) einbeziehen Geeignete Sprachen und Werkzeuge verwenden Die Spezifikation so früh wie möglich prüfen und dem Konfigurationsmanagement unterstellen Die Spezifikation intensiv verwenden

Vorgehensempfehlung

Tabelle 2.1

Abkürzungen

Typ	Kürz
class	-
package	pkg
component	cmp
deployment	dep
use case	uc
activity	act
state machine	stm
interaction	sd

Grundkonzepte

Modell

Ein Modell liefert ein eine Ansicht eines Systemes

1. Einn Modell versammelt eine hierarchisch strukturierte Menge von Elementen.
2. Die Menge von Elementen beschreibt auf einer abstrakten Ebene das System zu einem bestimmten Zweck, es zeigt nur eine Sicht auf das System.
3. Das Modell beschreibt das System aus dieser Sicht vollständig.

Kommentar

Rechteckt mit geknickter Ecke Kann mir >1 Modellelementen Verbunden werden

Die UML definiert folgende Sprachpakete, nach denen sich auch die Spezifikation untergliedert: Actions, Activities, Classes, Components, Deployments, General Behavior, Information Flows, Interactions, Models, Profiles, State Machines, Structures, UseCases, Templates.

Compliance Level von UML

Siehe Abb. 3.26 L0: Klassen, Pakete und Typen L1: Communication, BasicBehavior, BasicInteraktion, Interfaces L2: BehaviorStateMachines, Ports,

Ausdruck

Ist ein strukturierter Baum von Symbolen Der entweder in Präfix oder infix Schreibweise ist

prä: mal (plus (2, 7), 4)

inf: (2 + 7) * 4)

Randbedingungen / constraint

<constraint> ::= ‘{‘ [<name> ‘:’] <Booleanexpression> ‘ }’

Entwickler entwickelt Entwicklungsobjekt

|
XOR
|

Tester tester Testobjekt

Ein Entwickler kann nur entwickeln oder testen, nie beides Gleichzeitig Ist immer ein boolescher Ausdruck bsp: XOR

DatenTyp

Ein Datentyp wird durch das Rechteckssymbol mit dem ihn identifizierenden Schlüsselwort notiert, also «datatype» für einen Datentyp, «enumeration» oder «primitive» für die speziellen Datentypen „Primitiver Typ“

Wie eine Klasse kann ein Datentyp (mit Ausnahme des Primitiven Typs) Attribute und Operationen enthalten. Im Unterschied zu einem Objekt sind Datentypen allerdings identitätslos, das heißt, die Instanzen eines Datentyps sind gleich, wenn ihre Werte gleich sind. Instanzen einer Klasse, also Objekte, sind dagegen auch dann unterschiedlich, wenn sie für alle Attribute die gleichen Werte besitzen.

Primitive Typ {bool, int, string, n}

Anwendung ding : Integer[*] steht für eine nicht begrenzte Menge an Dingen

Aufzählungstyp

Sind diskrete Werte, wie die Ampel einer Ampel {rot, gelb, grün, rot-gelb}

UML-Schlüsselwörter

Wort	Bedeutung
------	-----------

beschreiben von optimalen Verhalten, trennung von notwenigem

Mit hilfe von include gemeinsames verhalten herausfaktoriesieren

Definition einer Methode

- Welcher Ausschnitt der UML ist für mein Projekt sinnvoll?
- Welche Stereotypen werden in meinem Projekt benötigt?
- Wie werden die einzelnen Entwicklungsschritte mit der UML unterstützt?
- Wie stehen die Entwicklungsprodukte miteinander in Verbindung?

Mittel zur Verfeinerung eines Use-Cases

- Use-Case-Beschreibungen, um weitere Informationen festzuhalten
- Aktivitätsdiagramme, um den Ablauf eines Use-Cases zu beschreiben
- Zustandsautomaten, um die Zustände des Systems während des Use-Case-Verlaufes zu modellieren
- Sequenzdiagramme, um die Interaktionen zwischen den Benutzern und dem System darzustellen
- Use-Case-Diagramme

UseCaseBeschreibung und Aktivitätsdiagramme zur präzisen Beschreibung eines UseCases aus. Zustandsautomaten verwenden wir in der Analyse eher zur detaillierten Beschreibung von Objekten oder zur Modellierung der Vor und Nach bedingungen der UseCases

Use-Case-Bedingung

- Der Name des Use-Cases
- Die Kurzbeschreibung (Die Erläuterung des Use-Cases in 1–2 Sätzen)

- Die Akteure
- Der fachliche Auslöser (Der Grund, warum der Benutzer den Use-Case ausführt)
- Der normale Ablauf
- Die alternativen Abläufe
- Die Vor- und Nachbedingungen

Use-Case-Diagramme

Als erstes erstellen wir ein Kontextdiagramm, das als Komponenten Diagramm erstellt wird 'Abb.4.2.png'

Beschrieben der Funktionalitäten

Aktivitätsdiagramme

In den Use Cases macht man sich über mögliche Abläufe Gedanken, mit Aktivitätsdiagrammen legt man diese genauer fest. Modelliert werden Aktionen, welche vom System alleine, von einem Akteur oder beiden durchgeführt werden. Um dies abzubilden greift man entweder auf Aktivitätsbereiche zurück oder man schreibt Akteure in Klammern über deren Aktionsnamen

Include und Exclude

Verwenden von Kommentaren

Erstellen Sie zu jeder Aktion eine kurze Beschreibung mit dem wichtigsten Inhalt unabhängig davon, ob Sie diese Aktion durch ein weiteres Diagramm verfeinern oder nicht. Das hilft Ihnen, wenn Sie die Verfeinerung nicht direkt durchführen, und es hilft den Lesern zu entscheiden, ob sie die Verfeinerung betrachten müssen oder nicht.

Use-Case-Diagramme

'Abb.4.7.png'

Zustandsautomaten in der Analyse

Use-Cases nicht nur mit Aktivitätsdiagrammen sondern auch mit Zustandsautomaten verfeinern

1. Use-Case durch Zustandsautomaten beschreiben
2. Kombination der entstandenen Zustandsautomaten

'Abb.4.8.png'

Das Begriffsmodell

Bei rein textuell und alphabetisch Glossar geht der Zusammenhang verloren Es bietet sich das Klassendiagramm an

- Klassen mit Attributen
- Assoziation & Aggregation
- Generalisierung

'Abb.4.7.png'

Modellierungsszenarien

Im Folgenden trennen wir dazu zwischen dem Senden und Empfangen von Nachrichten, dem Auslösen eines fachlichen Ablaufs und dem fachlichen Ablauf an sich (dem Ablauf eines UseCases). Hinzu kommen noch die Interaktionen eines Benutzers mit dem System im Rahmen eines fachlichen Ablaufs. Dies erreichen wir in den Modellierungsszenarien durch die Anwendung von zwei Prinzipien:

- Die Aufgaben werden als einzelne Aktivitäten modelliert.
- Der Zusammenhang wird im Wesentlichen durch den Austausch von Signalen modelliert.

Einen Überblick über die potenziellen Zusammenhänge gibt Abbildung 4.10.
'Abb.4.10.png'

Aufgaben von Systemschnittstellen

- Aufbau von Nachrichten bzw. Aussehen und die Übergabeparameter bei einem externen Operationsaufruf
- Rückmeldung des externen Systems über den korrekten Empfang einer Nachricht
- Fachlich über den Inhalt der gesendeten Information

'Abb.4.11.png' Die Persistierung wird zur Unterscheidung als Sterotyp gekennzeichnet

Sendende Schnittstelle

Enge Kopplung

Das Zusammenstellen der Nachricht hängt von externen Faktoren ab Falls es zu einem Timeout kommt Problem sonst OK 'Abb.4.12.png'

Lose Kopplung

Hier wird als Sterotyp <<Internes Ereignis>> gesetzt, im Schnittstellen Prozess und nicht im fachlichen Ablauf wird die Fehlerbehandlung durchgeführt

'Abb4.13.png'

Interaktion in Dialogen

Falls:

- Nutzer ändert angezeigt Werte
- Nutzer lost Aktion aus
- Externes System, liefert neue Daten
- Externes System, liefert neue Daten auf die registert wird

Modellierung von Services

fasst eine Menge von Diensten zusammen BSP Interaktion von Kundenverwaltung, Verkauf und Anmeldeung 'Abb.4.21'

Es ist möglich eine Serviceoperation als Schnittstellen Operation der Komponente zu realisieren Dadurch wird die zu realisierende Operation durch einer Aktivität Representiert 'Abb.4.22-23.png'

Modellierung der DTOs

Klassendiagramme werden benutzt um fachlich relevante Begriffe zu definieren

Detailtiefe Für die maximale Detaillierungstiefe gibt es eine relativ einfache Regel: Gehen Sie maximal so tief, bis Sie eine Aktion aus dem Aktivitätsdiagramm genau einer Komponente Ihres Systems zuordnen können.

Merkmale des Use-Cases	Empfohlene Notation	Referenz
Kruze klare Abläufe	Strukturierter Text	
Ablauf-, Schrittorientiert	Aktivitätsdiagramm	Kap. 13
Einfach Daten oder Entities	Kommunikationsdiagramm	Kap. 16
Komplexe Daten oder Entities	Sequenzdiagramm	Kap. 15
Kein typischer Ablauf (Abfolgen)	Zustandsautomat	Kap. 14
Use-Case bündelt viele Szenarien	Interaktionsübersichtdiagramm	Kap. 18