Anwendungsfälle und Anwendungsfalldiagramme

Udo Kelter

16.03.2003

Zusammenfassung dieses Lehrmoduls

Anwendungsfälle (*use cases*) und Anwendungsfalldiagramme modellieren die Interaktion eines Systems mit seinen Benutzern und andern externen Instanzen. Sie werden meist innerhalb objektorientierter Analysemethoden zusammen mit anderen Modellen eingesetzt. In der Beschreibung eines Anwendungsfalls werden einzelne Schritte identifiziert, in denen der Anwendungsfall abgearbeitet wird. Anwendungsfälle können einander benutzen und erweitern. Dieses Lehrmodul führt die entsprechenden Begriffe und UML-Notationen ein und gibt Hinweise, wie Anwendungsfälle entwickelt werden sollten.

Vorausgesetzte Lehrmodule:

obligatorisch:
- Objektorientierte Modellierung
- Systemanalyse und Systemmodellierung

Stoffumfang in Vorlesungsdoppelstunden: 1.0

Inhaltsverzeichnis

1 Einordnung

Anwendungsfälle bzw. Anwendungsfalldiagramme sind Funktionsmodelle bzw. Teile von Funktionsmodellen eines zu entwickelnden Systems.

Im Gegensatz zu anderen Funktionsmodellen wird hier nicht immer angestrebt, die Funktionalität eines Systems vollständig zu modellieren; stattdessen werden Anwendungsfälle bzw. Anwendungsfalldiagramme vielfach nur dazu eingesetzt, punktuell die "wichtigsten" Funktionalitäten des Systems darzustellen, oft in Verbindung mit Klassendiagrammen. Klassendiagramme modellieren bekanntlich ebenfalls die Funktionalität eines Systems, so daß beide Modelltypen einander ergänzen. Anwendungsfälle werden oft als Teil objektorientierter Entwicklungsmethoden angesehen; im Prinzip sind sie aber unabhängig von objektorientierten Methoden.

Anwendungsfälle und Anwendungsfalldiagramme gehen auf Jacobson [Ja+92] zurück und wurden später als Teil der UML standardisiert. Anwendungsfälle sind als Methode vielfach nur vage definiert worden; dies hat zu einer Vielzahl von nicht kompatiblen Definitionen und Vorstellungen davon geführt, was ein Anwendungsfall ist und wie man Anwendungsfälle systematisch entwickelt. [Co00] zeigt nicht weniger als 24 unterschiedliche Definitionen auf.

2 Grundbegriffe

2.1 Anwendungsfälle

Ein **Anwendungsfall** (*use case*) ist ein typischer, wesentlicher betrieblicher Vorgang im modellierten Unternehmen. Synonym mit Anwendungsfall werden verwendet: **Geschäftsprozeß**, **Nutzungsfall** und in manchen Quellen (aber nicht hier) Szenario. Anwendungsfälle kann man auf verschiedenen Abstraktionsstufen betrachten:

– *aus Sicht des Unternehmens*: Ein Anwendungsfall ist hier eine Sequenz von geschäftlichen Transaktionen (oder Aktivitäten oder

Aufgaben), die in ihrer Gesamtheit eine signifikante Leistung des Unternehmens gegenüber einem Kunden oder einer sonstigen externen Instanz realisiert. Beschrieben werden hier Arbeitsabläufe aus einer ergebnisorientierten Sicht. Auf dieses Verständnis paßt die Bezeichnung Geschäftsprozeß besonders gut.

– *aus der technischen Sicht eines Informationssystems*: Ein Anwendungsfall ist hier eine Sequenz von Transaktionen, die von einem Akteur im Dialog mit dem System ausgeführt werden, um für den Akteur einen meßbaren Nutzen zu erzielen. Transaktion ist hier (datenbank-) technisch zu verstehen, insb. werden Transaktionen ganz oder gar nicht wirksam.

Bei der Systemanalyse ist nur die erste der beiden vorstehenden Definitionen sinnvoll; wir setzen sie daher i.f. voraus.

Anwendungsfälle beschreiben nur das Außenverhalten des Systems, also *was* es leistet, nicht hingegen, *wie* es die Leistung erbringt. Nicht durch Anwendungsfälle modelliert werden systeminterne Funktionen. Generell sind Anwendungsfälle nicht dazu gedacht bzw. geeignet, eine funktionale Zerlegung eines Systems zu modellieren.

Bei der Analyse werden stets *komplette Arbeitsabläufe* modelliert, unabhängig davon, ob sie komplett oder nur teilweise durch ein Softwaresystem automatisiert werden; die Frage, welche Teile eines Arbeitsablaufs automatisiert werden, ist zum Zeitpunkt der Analyse typischerweise offen.

2.2 Akteure

Ein **Akteur** (*actor*) modelliert eine Systembenutzerrolle bzw. allgemeiner eine externe Instanz, die mit dem System interagiert[1]. Beispiele hierfür sind:

– eine natürliche Person; dies ist der häufigste Fall. Wenn eine Person mehrere Rollen spielt, muß jede Rolle durch einen eigenen Akteur modelliert werden.

[1] Akteure ähneln stark den Begrenzern in Datenflußdiagrammen, s. [DFD].

– eine Organisationseinheit
– ggf. ein externes System

Akteure sind *nicht* Teil des modellierten Systems, sie interagieren mit dem System.

Graphisch dargestellt werden Akteure meist als Strichmännchen (visuelle Stereotypisierung). Ebensogut können Akteure als Klasse mit dem textuellen Stereotyp ≪`actor`≫ dargestellt werden.

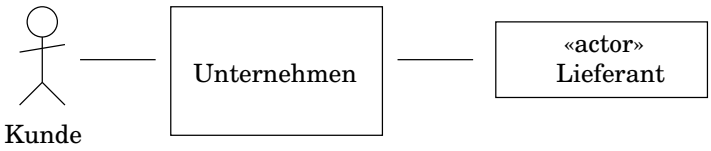

Abbildung 1: Modell eines Unternehmens

Ob eine Instanz Teil des Systems ist oder nicht, hängt natürlich auch von der Perspektive ab. Betrachtet man als System ein komplettes Unternehmen, so wären Kunden und Lieferanten externe Instanzen und würde als Akteure modelliert werden; die Buchhaltung hingegen wäre eine interne Instanz und würde nicht modelliert (s. Bild 1).

Innerhalb des Unternehmens ergibt sich eine andere Perspektive: aus Sicht der Bestellungsannahme ist die Buchhaltung eine externe Instanz und muß als Akteure modelliert werden (s. Bild 2).

Wenn mehrere einander ähnliche Akteure vorhanden sind, kann man eine Vererbungshierarchie bilden - in diesem Sinne sind Akteure ganz normale Klassen. Graphisch dargestellt wird die Vererbungsbeziehung zwischen zwei Akteuren durch den gleichen Pfeil wie bei Klassen.

2.3 Anwendungsfalldiagramme

Im allgemeinen deckt ein System mehrere Anwendungsfälle ab, wobei teilweise die gleichen Akteure involviert sein können. Ein **Anwendungsfalldiagramm** beschreibt das Zusammenspiel der Anwen-

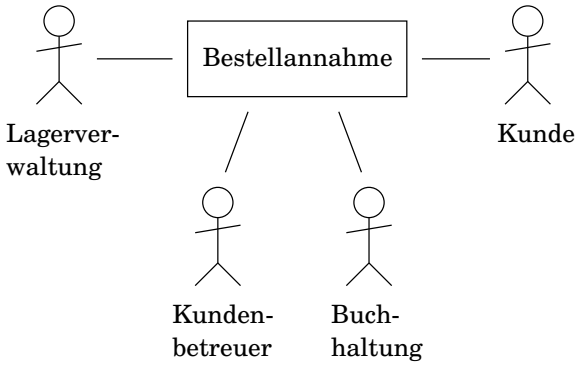

Abbildung 2: Modell eines Unternehmensteils

dungsfälle untereinander und mit den Akteuren. Sinn und Zweck eines Anwendungsfalldiagramm liegen vor allem darin, einen Überblick über die Anwendungsfälle zu gewinnen; fast alle Details der Beschreibung eines Anwendungsfalls fehlen im Diagramm und müssen einem separaten Dokument entnommen werden.

Graphisch dargestellt wird ein Anwendungsfall durch ein Oval, in dem der Name des Anwendungsfalls steht. Die Ovale stehen in einer durch ein Rechteck gebildeten Fläche, die das modellierte System symbolisiert; die Akteure stehen außerhalb dieser Fläche. Bild 3 zeigt ein Beispiel. Die Beteiligung eines Akteurs an einem Anwendungsfall wird durch eine Linie zwischen den beiden Symbolen dargestellt.

3 Anwendungsfälle

3.1 Spezifikation von Anwendungsfällen

Die UML läßt es offen, wie ein Anwendungsfall textuell beschrieben wird[2]; üblich sind textuelle Beschreibungen. Die folgende Liste enthält

[2] s. [UML99], 3.54.1: "The behavior of a use case can be described in several different ways, depending on what is convenient: often plain text is used, ..."

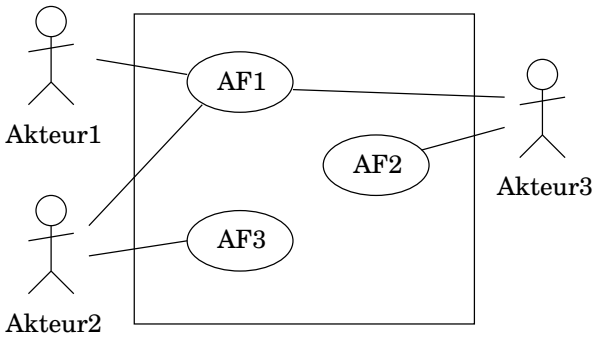

Abbildung 3: Anwendungsfalldiagramm

die typischen Beschreibungsmerkmale eines Anwendungsfalls:

Nummer des Anwendungsfalls: eindeutige Nummer

Name: eindeutiger Name

Systemumfang: z.B. ganzes Unternehmen, eine bestimmte Abteilung oder ein konkretes Softwaresystem (s. Abschnitt 5)

beteiligte Akteure: Diese Angabe ist im Prinzip redundant, da man die beteiligten Rollen auch im Anwendungsfalldiagramm erkennen kann.

Ziel: kurze verbale Beschreibung der Zielsetzung des Anwendungsfalls bzw. Geschäftsprozesses unter der Annahme, daß der Anwendungsfall erfolgreich verläuft. Beispiele für Ziele sind: “eine Bestellung aufnehmen”, “Geld aus dem Automaten abheben”, “ein Buch in der Bibliothek ausleihen”.

Ziel-Klassifizierung: s. Abschnitt 5 (Detaillierungsgrad des Ziels)

Kategorie: primär, sekundär oder optional. Die Bedeutungen sind:

- primär: häufiger Fall, sollte besonders gut benutzbar sein
- sekundär: weniger häufiger Fall, Unterstützung dieses Falls dennoch unverzichtbar
- optional: Unterstützung dieses Falls ist nützlich, aber verzichtbar

Vorbedingung: Bedingungen an den Zustand des Systems, bevor der

Anwendungsfall eintreten kann

Nachbedingung im Erfolgsfall: Zustand des Systems, nachdem der Anwendungsfall erfolgreich durchlaufen wurde, i.w. also das betriebliche Ergebnis.

Die Nachbedingung kann die Vorbedingung eines anderen Anwendungsfalls sein, d.h. durch entsprechende Vor- und Nachbedingungen können implizit Reihenfolgerestriktionen zwischen Anwendungsfällen festgelegt werden.

Ausnahmen und Fehlersituationen: Beschreibung der möglichen Ausnahmen und Fehlersituationen aus fachlicher (nicht technischer) Sicht. Beispielsweise könnte bei der Aufnahme einer Bestellung festgestellt werden, daß die Ware nicht lieferbar ist.

Nachbedingung im Fehlerfall: Zustand des Systems, nachdem der Anwendungsfall nicht erfolgreich durchlaufen wurde.

auslösendes Ereignis: Ereignis, das den Geschäftsvorgang initiiert

Ablaufbeschreibung: Hier unterstellen wir, daß die Bearbeitung des Anwendungsfall in mehrere Schritte oder Aktionen gegliedert werden kann. Angegeben wird die Folge dieser Schritte.

Der Begriff "Folge" ist hier nicht streng wörtlich zu verstehen, einzelne Schritte können z.B. auch parallel ausgeführt werden. Anwendungsfälle sind nicht dazu gedacht (und auch nicht dazu geeignet), detailliert Ablaufstrukturen zu beschreiben; hierzu sollten bei Bedarf zusätzliche Aktivitäts-, Sequenz-, Kollaborations- oder Zustandsübergangsdiagramme benutzt werden. Es geht hier in erster Linie nur darum, die einzelnen Schritte zu identifizieren und in einen Kontext zu setzen.

Beschrieben wird hier nur der am häufigsten auftretende Standardfall; Sonderfälle werden separat beschrieben.

Die Zahl der Schritte sollte überschaubar bleiben; ggf. sind gröbere Schritte zu bilden, die separat verfeinert werden.

Varianten/Alternativen: Hier handelt es sich um Variationen des Anwendungsfalls, bei denen vorhandene Schritte modifiziert werden, der Ablauf aber im Prinzip erhalten bleibt. Beispielsweise kann es bei einer Zahlung mehrere alternative Zahlungswege geben.

Erweiterungen: Hier sind Fälle zu beschreiben, die zu Abweichungen vom Normalablauf führen, die auch neue Einzelschritte beinhalten können.

nichtfunktionale Anforderungen: z.B. Voraussetzungen, die die Plattformen oder Arbeitsumgebung erfüllen muß, erwartete Antwortzeiten oder Durchsatz, Prioritäten usw.

Ansprechpartner und Sitzungen: Beschreibung der Personen / Rollen, die an der Definition des Anwendungsfalls beteiligt waren, ggf. Verweise auf Sitzungsprotokolle

Anmerkungen: Hier können z.B. wichtige Entwurfsentscheidungen dokumentiert werden.

Dokumente, Referenzen, Dialogbeispiele: Hier können Materialien aufgelistet werden, die den Anwendungsfall illustrieren oder eine Basis für seine Definition darstellen.

ergänzende Diagramme: Diese können z.B. einzelne Szenarios (s.u.) oder systeminterne Zustände und Abhängigkeiten genauer spezifizieren.

In der Literatur sind diverse ähnliche Schemata zur Beschreibung von Anwendungsfällen vorgeschlagen worden (unter Bezeichnungen wie Geschäftsprozeßschablone oder *use case template* [Ba99, Co00]). Man kann in konkreten Fällen natürlich irrelevante Beschreibungsmerkmale weglassen.

3.2 Szenario vs. Anwendungsfall

Zentrales Orientierungsmerkmal eines Anwendungsfalls ist das Ziel, das die beteiligten Akteure bei der Interaktion mit dem System erreichen wollen. In einen Anwendungsfall ist daher nur das aufzunehmen, was zum Erreichen des Ziels oder für die Feststellung, daß das Ziel nicht erreichbar und der Geschäftsvorgang abzubrechen ist, relevant ist. Je nach den Bedingungen, die bei der Verfolgung des Ziels aufgetreten sind, sind verschiedene Abläufe denkbar. Ein einzelner derartiger Ablauf wird **Szenario** genannt. Ein Szenario wird durch folgende Merkmale charakterisiert:

- das verfolgte Ziel
- die unterstellten Bedingungen
- das erreichte Resultat, also Erfolg oder Fehlschlag, und ggf. zusätzliche Detailangaben

Ein Anwendungsfall entspricht in diesem Sinne einer Menge von Szenarios, die das gleiche Ziel, aber verschiedene Bedingungen haben.

Im oben vorgestellten Beschreibungsschema für Anwendungsfälle erscheinen die Szenarios nur versteckt, werden also nicht explizit einzeln dargestellt. Manchmal ist es aber sinnvoll, ein einzelnes Szenario separat darzustellen, wobei vor allem Sequenz- und Kollaborationsdiagramme infrage kommen. Alle Szenarios einzeln zu behandeln scheitert i.d.R. an der kombinatorischen Explosion der Zahl der Szenarios. Beispielsweise könnte bei einem Kaufvorgang der Kunde bar, mit Kreditkarte, mit EC-Karte oder per Lastschrift bezahlen und die Ware könnte per Post, per Paketdienst, per Spedition oder durch Selbstabholung geliefert werden. Derartige Fälle sollten durch Varianten einzelner Schritte behandelt werden.

3.3 Ein Beispiel

Als Beispiel betrachten wir den Fall, daß Sie bei einem Besuch bei Bekannten eine Tasse Kaffee über den schönen neuen Multimedia-Rechner geschüttet haben und daß ausgerechnet das Medium Wasser von diesem Rechner nicht unterstützt wird. Wir betrachten diesen Anwendungsfall "Abwicklung Haftpflichtschaden" aus Sicht Ihrer Haftpflichtversicherung. Sie, Ihre Bekannten und ggf. zusätzlich ein Gutachter sind die externen Akteure.

Name: **Abwicklung Haftpflichtschaden**
Systemumfang: Abteilung Haftpflichtschäden
beteiligte Akteure: Versicherungsnehmer, Geschädigter, Gutachter
Ziel: Schadenersatzzahlung an Geschädigten
Ziel-Klassifizierung: summarisches Ziel
Kategorie: primär

Vorbedingung: Schadensmeldung durch Versicherungsnehmer liegt vor

Nachbedingung im Erfolgsfall: Schadenersatz wurde an den Geschädigten überwiesen

Ausnahmen und Fehlersituationen:

1. keine gültige Police des Versicherungsnehmers
2. vorliegendes Schadensereignis nicht / nur teilweise abgedeckt

Nachbedingung im Fehlerfall: bei Ausnahme

1. Information an Geschädigten, daß kein Schadenersatz geleistet wird; Abschluß und Archivierung des Falls
2. Information an Geschädigten, in welchem Umfang Schadenersatz geleistet wird und Schadenersatzzahlung an Geschädigten; Abschluß und Archivierung des Falls

auslösendes Ereignis: Schadensmeldung durch Versicherungsnehmer

Ablaufbeschreibung:

1 Schadensmeldung incl. Beweisunterlagen erfassen

2 Versicherung prüft Formalia (ist Schadensrisiko im Rahmen der Versicherung versichert? bestehen Beitragsrückstände?)

3 Versicherungsmitarbeiter prüft Schaden und stellt Höhe des Schadenersatzes fest

4 Versicherung zahlt Schadenersatz an Geschädigten

Erweiterungen:

Erweiterungen von Schritt 3:

3a zusätzliche Ortsbesichtigung erforderlich; Schritte:

3a1 Versicherungsmitarbeiter stimmt Ortsbesichtigungstermin mit Geschädigtem und ggf. Versicherungsnehmer ab

3a2 Ortsbesichtigung

3a3 Versicherungsmitarbeiter fertigt Protokoll über Ortsbesichtigung an, archiviert Fotos und andere Unterlagen

3b zusätzliches Wertgutachten durch externen Gutachter (Schadenssachverständiger) erforderlich:

3b1 anhand Gutachterkartei prüfen, welche Gutachter verfügbar sind, einen auswählen und beauftragen

3b2 Gutachten erfassen und archivieren

3b3 Gutachterhonorar zahlen

Varianten/Alternativen: zu Schritt 4:

- Zahlung durch Verrechnungsscheck
- Zahlung durch Überweisung auf Girokonto
- Zahlung durch Postanweisung

Bei der vorstehenden Gliederung der Abläufe wurde davon ausgegangen, daß eine Ortsbesichtigung und die Einschaltung eines externen Schadenssachverständigen im Normalfall nicht erforderlich sind. Wäre eine Ortsbesichtigung der Normalfall, würde man sie schon im "Normalablauf" aufführen.

4 Beziehungen zwischen Anwendungsfällen

Zwei Anwendungsfälle können durch folgende Beziehungen zusammenhängen:

- durch eine Vererbungsbeziehung, die mit dem üblichen Vererbungspfeil dargestellt wird. Der der Oberklasse entsprechende Anwendungsfall kann abstrakt (im Sinne von nicht instanziierbar) sein. Die Unterklasse erbt hier insb. die Akteure und das abstrakte Verhalten des abstrakteren Anwendungsfalls.
- Eine benutzt-Beziehung vom Typ `includes`; diese wird durch den üblichen benutzt-Pfeil (gestrichelte Linie, offene Pfeilspitze) und das Stereotyp ≪`includes`≫ dargestellt. Eine solche Beziehung entspricht einer Art Unterprogrammaufruf, d.h. der eingefügte Anwendungsfall ist Teil des Ablaufs des einfügenden Anwendungsfalls.

  Typischerweise entstehen solche Beziehungen dadurch, daß man in mehreren Anwendungsfällen gleiche Untersequenzen findet, diese als eigenen Anwendungsfall herausfaktorisiert und dann in den ursprünglichen Anwendungsfällen aufruft; in diesen wird dann

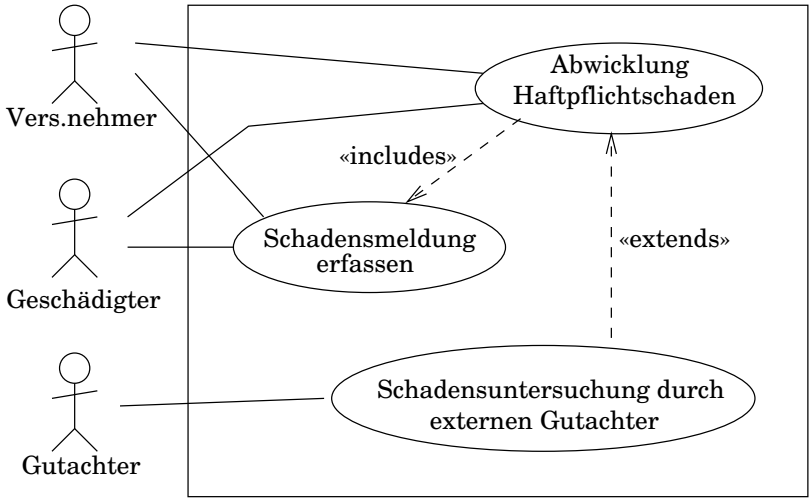

Abbildung 4: Beziehungen zwischen Anwendungsfällen

genau ein Arbeitsschritt auftreten, der den kompletten Unter-Anwendungsfall repräsentiert.

Beispielsweise kann es bei ganz unterschiedlichen Anwendungsfällen nötig sein, die Adresse eines Kunden zu korrigieren bzw. neu zu erfassen; dies wird naheliegenderweise ein benutzter separater Anwendungsfall sein. Dieses Beispiel zeigt zugleich, daß ein benutzter Anwendungsfall auch unabhängig von benutzenden Anwendungsfällen existieren kann.

– Eine benutzt-Beziehung vom Typ `extends`; diese wird durch den üblichen benutzt-Pfeil und das Stereotyp ≪`extends`≫ dargestellt. Der erweiternde Anwendungsfall wird in den erweiterten eingefügt. Die Erweiterung besteht typischerweise aus einer Behandlung von Ausnahme-, Fehler- und Sonderfällen. Ein Beispiel für einen erweiternden Anwendungsfall könnte “Schadensuntersuchung durch externen Gutachter” sein, der den Anwendungsfall “Abwicklung Haftpflichtschaden” aus Abschnitt 3.3 erweitert.

Durch weitere UML-Konstrukte, die allerdings nicht präzise beschrieben sind und die wir hier nicht wiedergeben, können zusätzlich die Gelegenheiten angegeben werden, zu denen ein oder mehrere eingefügte oder erweiternde Anwendungsfälle aufgerufen werden.

Der Unterschied zwischen Einfügung und Erweiterung (A `extends` B vs. B `includes` A) besteht darin, daß die Einfügung obligatorisch ist, während die Erweiterung optional ist. Diese Unterscheidung ist aber oft unscharf und in ersten Versionen der Diagramme manchmal nicht bestimmbar.

5 Klassifizierung von Anwendungsfällen

Aus den vorstehenden Beispielen sollte bereits klargeworden sein, daß Anwendungsfälle auf verschiedenen Abstraktionsniveaus auftreten und betrachtet werden können. Dies ist insofern sehr nachteilig, als dadurch bei der Systemanalyse zu viele Aspekte zugleich und zur unpassenden Zeit behandelt werden. Zur Lösung dieses Problems schlägt [Co00] einige Merkmale vor, anhand derer Anwendungsfälle bzgl. ihres Abstraktionsgrads klassifiziert werden können. Der Nutzen dieser Klassifizierungsmerkmale liegt darin, daß Diskussionen zwischen Analytikern und dem Kunden besser strukturiert und die Anwendungsfälle anhand ihres Abstraktionsgrads gruppiert werden können.

Die Werte dieser Klassifizierungsmerkmale sollten bei der Spezifikation eines Anwendungsfalls angegeben werden (vgl. Beschreibungsschema in Abschnitt 3.1).

I.f. stellen wir die einzelnen Klassifizierungsmerkmale vor.

Systemumfang. Hier dreht es sich um die Frage, wo die Grenzen des betrachteten Systems liegen. Wie wir schon früher gesehen haben, beeinflußt dies die Entscheidung, welche Instanzen als extern anzusehen sind und demzufolge als Akteure zu modellieren sind.

Ein denkbarer Fall ist die “Systemebene”, hier wird genau das zu entwickelnde System betrachtet. Ein zweiter Fall ist die “Unternehmensebene”; hier sieht man das ganze Unternehmen incl. anderer Softwaresysteme und manueller Verarbeitungskapazitäten als System

an. Denkbar sind auch Zwischenstufen, bei denen man einzelne Bereiche des Unternehmens oder die gesamte EDV als System ansieht. Auf der Unternehmensebene wird man eher strategische Ziele in den Anwendungsfällen vorfinden, auf der Systemebene können auch relativ detaillierte Ziele auftreten.

Detaillierungsgrad des Ziels. Ein Beispiel eines Anwendungsfalls ist ein Verkauf von Waren über eine WWW-Schnittstelle. Ziel des Anwendungsfalls ist der Verkauf eines Produkts, ein Detailziel ist die Unterstützung des Kunden bei der Suche nach einem geeigneten Produkt und die Angabe detaillierter Informationen über ein Produkt, sofern der Kunde danach verlangt. Der Verkauf von Waren ist ein sehr generelles Ziel, das ggf. je nach Art der Ware noch spezialisiert werden muß. Eine Suchfunktion realisiert nur ein Teilziel, das innerhalb verschiedener Anwendungsfälle auftreten kann - mit einiger Wahrscheinlichkeit werden derartige Funktionen als mehrfach benutzte Anwendungsfälle modelliert. Sofern man Anwendungsfälle durch ≪`includes`≫ -Beziehungen strukturiert, werden die Ziele der benutzten Anwendungsfälle automatisch zu "Etappenzielen" der benutzenden Anwendungsfälle. [Co00] unterscheidet dementsprechend drei Detaillierungsstufen der Ziele:

- **summarische Ziele** (z.B. allgemein der Verkauf von Produkten oder die Abwicklung von Garantiefällen)
- **Benutzerziele** (z.B. Verkauf eines Produkts aus einer konkreten Warengruppe)
- **Teilziele** (z.B. die Suche innerhalb von Produktbeschreibungen)

Detaillierungsgrad der Interaktion. Eine dritte Dimension ist der Detaillierungsgrad, auf dem die Interaktion zwischen Akteuren und dem modellierten System betrachtet wird. Hier kann man ganz grob eine technische und eine inhaltliche Ebene unterscheiden. Auf der technischen Ebene könnte beschrieben werden, daß man mit der Tab-Taste in das nächste Eingabefeld kommt oder wie Formulare bei

einer umfangreichen Datenerfassung aufgebaut sind. Auf der inhaltlichen Ebene interessieren solche Details nicht, man würde hier eher davon reden, daß die Adreßdaten erfaßt werden oder daß ein Vorgang im Schriftverkehr identifiziert wird.

Durch GUI-Prototypen, die vielfach als Ergänzung zu Anwendungsfällen angefertigt werden, wird eher die technische Ebene angesprochen.

Die Erfahrung zeigt, daß man sich besser auf die inhaltliche Ebene konzentriert und technische Details nur in Ausnahmefällen in den Anforderungsdokumenten festlegt, in Anwendungsfällen also i.d.R. keine technischen Aspekte der Interaktion behandelt. Dies steht nur scheinbar im Gegensatz dazu, daß sich GUI-Prototypen in der Praxis als sehr nützlich bewährt haben. GUI-Prototypen sollte man als Wegwerf-Prototypen ansehen, denn die Gestaltung von GUIs muß oft, wenn der fachliche Kern eines System konsolidiert worden ist, noch einmal überarbeitet werden. Ggf. sind auch die ersten GUI-Entwürfe ergonomisch nicht ausgereift.

6 Vorgehen beim Entwickeln von Anwendungsfällen

Anwendungsfälle können als Analysemethode sehr früh eingesetzt werden. Als Funktionsmodelle kommen sie dem Denkverhalten vieler Anwender, die eher in Abläufen und Interaktionen denken, entgegen.

Anwendungsfälle sind in der hier vorgestellten Form unabhängig von einem Datenmodell, können also zeitlich vor einem Datenmodell (ER- oder Klassendiagramm) entwickelt werden.

Wenn man pro Anwendungsfall eine Seite Text veranschlagt und bei kleinen bis mittleren Systemen von ca. 20 bis 100 Anwendungsfällen ausgeht, ist der Arbeitsaufwand zur Notation der Anwendungsfälle durchaus signifikant. Ihr Entwicklungsprozeß sollte in etwa wie folgt strukturiert werden:

1. Zunächst wird eine initiale Liste von Anwendungsfällen erstellt; die Fälle werden hier nur sehr kurz beschrieben, i.w. nur durch Angabe

der Akteure, des Ziels und der Klassifizierungsmerkmale

2. Es wird versucht, Beziehungen zwischen den Anwendungsfällen zu identifizieren, ggf. gemeinsam benutzte Teile herauszufiltern und als weitere Anwendungsfälle hinzuzufügen; hierbei werden Anwendungsfalldiagramme erstellt. Ergebnis ist eine (halbwegs) konsolidierte Liste von Anwendungsfällen.
3. In den einzelnen Anwendungsfällen wird jetzt der Ablauf im Normalfall ausgearbeitet und dabei ggf. Schritt 2 wiederholt.
4. In den einzelnen Anwendungsfällen werden die Varianten und Sonderfälle ausgearbeitet.
5. [Co00] schlägt vor, zum Schluß noch die Schnittstellen, über die mit den Akteuren kommuniziert wird (z.B. interaktive Schnittstellen, Dateien, Datenbanken), zu notieren.

Literatur

[Ba99] Balzert, Heide: Lehrbuch der Objektmodellierung; Spektrum Akademischer Verlag; 1999

[Co00] Cockburn, Alistair: Structuring use cases with goals; http://-members.aol.com/acockburn/papers/usecases.htm; 2000/02

[Ja+92] Jacobson, I.; Christerson, M.; Jinsson, P.; Övergaard, G.: Object-oriented software engineering - a use case driven approach; Addsion Wesley; 1992

[UML99] OMG Unified Modeling Language Specification (draft, Version 1.3 alpha R5, March 1999); OMG; 1999

[DFD] Kelter, U.: Lehrmodul "Funktionsmodellierung mit Datenflußdiagrammen"; 2000/04

Glossar

Akteur (*actor*): externe Instanz (Person, Geräte usw.), die mit dem modellierten System interagiert

Anwendungsfall (*use case*): beschreibt eine typischen Ablauf, wie ein modelliertes System arbeitet bzw. aus der externen Sicht von Akteuren genutzt wird; kann in verschiedenen Abstraktionsstufen eingesetzt werden; beschreibt ein oder mehrere Szenarien

Anwendungsfalldiagramm, auch **Geschäftsprozeßdiagramm** (*use case diagram*): Diagramm, das stark vereinfacht dargestellte Anwendungsfälle, deren Beziehungen sowie die involvierten Akteure darstellt (ooa)

extends-Beziehung zwischen Anwendungsfällen**:** drückt aus, daß der erweiternde Anwendungsfall den anderen um Ausnahme-, Fehler-, Sonderfällen o.ä. erweitert

includes-Beziehung zwischen Anwendungsfällen**:** drückt eine Benutzung ähnlich einem Unterprogrammaufruf aus

Szenario (*scenario*): Folge von Ablaufschritten, die unter ganz bestimmten Bedingungen durchlaufen werden; ohne Varianten oder Fallunterscheidungen

Index