

Anforderungsanalyse

Anforderungen und Anwendungsfälle

In dieser Vorlesung befassen wir uns mit der Frage, wie wir die Anforderungen an das zu konstruierende System (im Folgenden kurz *kS*) finden und beschreiben.

Man kann zwei Arten von Anforderungen unterscheiden:

Funktionale Anforderungen beschreiben, wofür das *kS* geeignet sein soll, welchen Zweck es erfüllen soll und welchen Funktionen es demzufolge ausführen muss. Funktionale Anforderungen sind als diejenigen, die den Zweck betreffen dessentwegen die Software überhaupt gebaut wird.

Nichtfunktionale Anforderungen betreffen Qualitätsmerkmale des *kS*, wie

- Benutzbarkeit
- Verfügbarkeit
- Leistungsfähigkeit
- Sicherheit
- Änderbarkeit
- Testbarkeit

Wichtig ist dabei stets der Bezug der Anforderungen auf den letztlich zu erreichenden Zweck des Systems, das „Geschäftsziel“.

Ziel

- Festlegung der Funktionalität des *kS*
- Überprüfen der Konsistenz und Vollständigkeit dieser Festlegung
- Formulierung der qualitativen (nichtfunktionalen) Eigenschaften des *kS*

Vorgehen

Um die funktionalen Anforderungen zu finden, kann man eine *Aufgabenanalyse* durchführen: Man formuliert die *Ziele*, *Aufgaben* und *Aktionen*, die aus Sicht der Benutzer des *kS* zu berücksichtigen sind.

Ziele beschreiben, was erreicht werden soll, und zwar unabhängig davon wie das Ziel erreicht werden kann. Die Ziele ergeben sich aus dem Anliegen der Benutzer.

Aufgaben sind erforderliche Schritte, um ein Ziel zu erreichen. Auch hier ist die Sache im Vordergrund: es geht um die Ermittlung der Schritte, die durch das jeweilige Anwendungsgebiet erforderlich sind, um ein Ziel zu erreichen.

Aktionen werden durch das Verfeinern der Aufgaben beschrieben, hier kommt das *kS* ins Spiel – wir beschreiben, welche Aktionen mit dem *kS* ausgeführt werden sollen, um das Ziel zu erreichen.

In der Aufgabenanalyse kann man mit verschiedenen Techniken die Informationen aus dem Anwendungsgebiet und über das *kS* erschließen:

- Interviews, Diskussion mit Anwendern
- Brainstorming
- Ergebnisse der Domänenanalyse (insbesondere alle Informationen über den Ist-Zustand)
- Vergleich mit Referenzmodellen
- Prototyping

Eine weit verbreitete Art und Weise, die funktionalen Anforderungen zu erfassen, sind Anwendungsfälle, die im folgenden Abschnitt diskutiert werden.

Nichtfunktionale Anforderungen kann man in Form von Listen oder auch (besser) von Qualitätsszenarien festhalten.

Anwendungsfälle

Ein Anwendungsfall beschreibt, wie ein Anwender eines seiner Ziele durch das Verwenden des Systems erreicht.

A key attitude in use case work is to focus on the question „How can using the system provide observable value to the users or fulfill their goals?“, rather than merely thinking of system requirements in terms of a „laundry list“ of features or functions

– Craig Larman

So beschreibt ein Autor (etwas zugespitzt) die Perspektive der Anwendungsfälle: eine Interaktion mit dem *kS*, aus dem ein Nutzen für den Anwender entsteht, mit dem er eines seiner Ziele oder Teilziele erreicht. Anwendungsfälle werden also nicht aus der Sicht des *kS*, als Schritte oder Aufgaben eines Ablaufs im *kS* formuliert, sondern stets aus der Sicht des Anwenders.

Anwendungsfall	UC123
Name	Artikel in den Warenkorb legen
Ziel	Der Benutzer entscheidet sich, ein Buch seiner Bestellung hinzuzufügen.
Akteur	Kunde
Auslöser	Button „In den Warenkorb legen“ auf der Seite Buchdetails
Vorbedingung	keine
Nachbedingung (Erfolg)	Artikel ist im Warenkorb verzeichnet
Nachbedingung (Misserfolg)	Artikel ist nicht im Warenkorb verzeichnet Benutzer ist über den Grund informiert
Standardablauf	1. Der Benutzer klickt auf der Seite „Buchdetails“ den Button „In den Warenkorb legen“
	2. Das System überprüft, ob das Buch noch verfügbar ist und fügt dann dem Warenkorb eine neue Position mit dem Buch und Stückzahl 1 hinzu. Das System reduziert den Bestand um 1.
	3. Der Benutzer kann jetzt auf der Seite „Warenkorb editieren“ das Ergebnis sehen und ggfs. die Stückzahl verändern
Alternativabläufe	Das Buch ist nicht mehr verfügbar ...
	Das Buch ist bereits im Warenkorb ...

Abbildung 1: Anwendungsfall „Artikel in den Warenkorb legen“

Anwendungsfälle werden in der Regel in tabellarischer Form beschrieben. (Dabei werden von verschiedenen Autoren unterschiedliche Gliederungen vorgeschlagen – am Besten, man einigt sich in einem Projekt auf eine Gliederung, die zur jeweiligen Art des Projekts passt.)

Ein beispielhafte, unvollständiger Ausschnitt eines Anwendungsfalls bei der Bestellung in einer Internet-Buchhandlung sieht etwa aus wie in [Abb. 1](#)

Eine Alternative zur Darstellung des Ablaufs in einem Anwendungsfall durch einen Text, der die Schritte darstellt, wie in obigem Beispiel, kann man auch das Aktivitätendiagramm einsetzen

Aktivitätendiagramm der UML

Im Beispiel des obigen Anwendungsfalls könnte ein Aktivitätendiagramm etwa so aussehen wie in Abb. 2

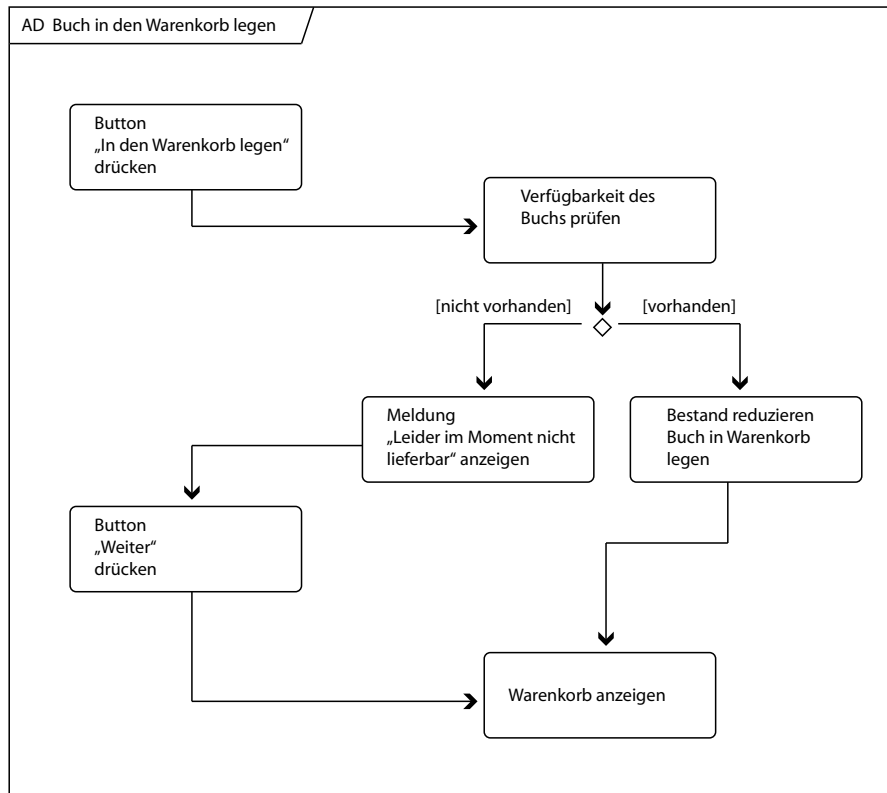


Abbildung 2: Aktivitätendiagramm zum Anwendungsfall „Buch in den Warenkorb legen“

Das Aktivitätendiagramm hat viele Elemente, die wichtigsten davon sind:

Schwimmbahnen Man verzeichnet Akteure im Aktivitätendiagramm, die für die Ausführung von Aktivitäten verantwortlich sind, jeder von ihnen erhält eine sogenannten Schwimmbahn, in die die Aktivitäten in seiner Obhut eingezeichnet werden.

Darstellung eines Datenflusses Zusätzlich zum reinen Ablauf in einem Aktivitätendiagramm können auch Objekte (Datencontainer) eingezeichnet werden, die mit dem Ablauf von Aktivität zu Aktivität weitergereicht werden.

Subaktivitäten Aktivitäten in einem Aktivitätendiagramm können selbst komplex sein und durch ein eigenes Aktivitätendiagramm beschrieben werden, das eine Verfeinerung darstellt. Auf diese Weise können komplexe Abläufe hierarchisch gegliedert werden und sind damit übersichtlicher.

Verzweigung und Verschmelzung Eine Verzweigung teilt einen Ablauf in unabhängige Zweige auf, die parallel verarbeitet werden können, eine Verschmelzung fügt die Zweige wieder zu einem gemeinsamen Kontrollfluss zusammen.

Darüber hinaus gibt es viele weitere Möglichkeiten im Aktivitätendiagramm, die man zum Beispiel im Buch von Harald Störrle, UML2 für Studenten, findet.

Anwendungsfalldiagramm

Die Anwendungsfälle kann man in einer Liste verzeichnen oder auch in einem Anwendungsfalldiagramm. Die Elemente im Anwendungsfalldiagramm sind

- Systemgrenze
- Akteure
- Anwendungsfälle
- Interaktion zwischen Akteur und Anwendungsfall
- Beziehungen zwischen Anwendungsfällen
 - Substituierbarkeit („kann an Stelle von ... treten“)
 - `<<include>>` (Inkludierter Anwendungsfall wird als Teil des Ablaufs des Anwendungsfalls benutzt)
 - `<<extends>>` (Anwendungsfall kann optional unseren Anwendungsfall erweitern)

Beispielhaft zeigt Abbildung 3 die Elemente im Anwendungsfalldiagramm.

Bemerkungen:

1. Das Anwendungsfalldiagramm wird in seiner Aussagekraft oft überschätzt – oder auch für falsche Zwecke eingesetzt. Es ist nicht viel mehr als eine strukturierte Übersicht über die Liste der Anwendungsfälle – etwas anschaulicher als eine reine Liste. Man sollte es nicht dazu verwenden, um Abläufe selbst darzustellen!

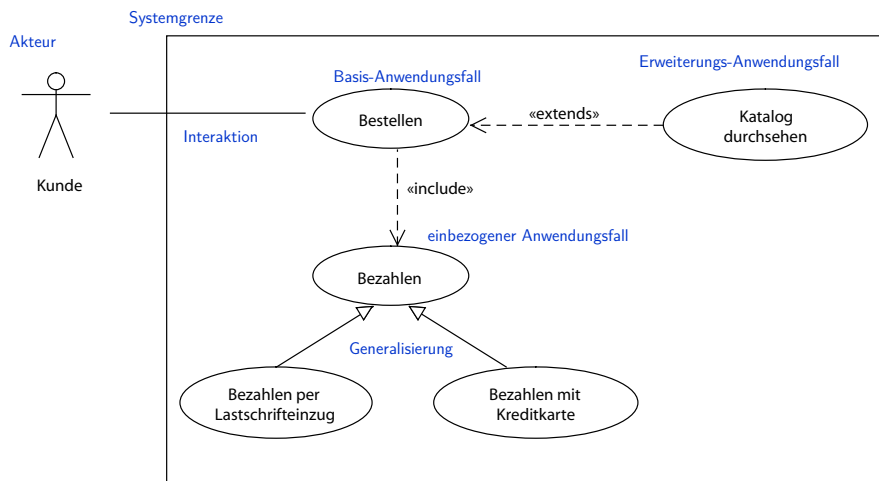


Abbildung 3: Elemente im Anwendungsfalldiagramm

2. Nicht in jedem Projekt können die Anforderungen vollständig mit Anwendungsfällen erfasst werden. In manchen Branchen sind andere Vorgehensweisen und Werkzeuge (z.B. DOORS von Telelogic) verbreitet, in denen Anwendungsfälle Teilelement sind.
3. Die Frage, wie man *systematisch* und *vollständig* Anforderungen mit Anwendungsfällen erfasst, bleibt weitgehend offen – die Flexibilität beim Aufstellen von Anwendungsfällen kann leicht mit dem Anspruch auf Konsistenz bei der Anforderungsanalyse in Konflikt geraten.
siehe dazu z.B. die Diskussion von Stephen Ferg http://www.jacksonworkbench.co.uk/stevefergspages/papers/ferg--whats_wrong_with_use_cases.html

Benutzeroberfläche und Interaktionsdesign

Viele Softwaresysteme haben heute eine graphische Benutzeroberfläche, die den Anwendern vielfältige und komfortable Möglichkeiten der Interaktion mit dem System geben. Umso wichtiger ist es, die Benutzeroberfläche sorgfältig zu gestalten.

Aber auch bei Systemen ohne graphische Benutzeroberfläche sollte man die Gestaltung der Interaktion mit Anwendern (oder anderen Systemen) nicht vernachlässigen.

Eine wichtige Rolle spielt die Diskussion über die Benutzeroberfläche für das Überprüfen der gefundenen Anforderungen, namentlich der Anwendungsfälle: mit einem Prototypen der Benutzeroberfläche können sie

durchgespielt werden.

Ziel

- Ein „Bild“ des laufenden Systems für die Analyse und Diskussion von Anforderungen
- Anschauliche Darstellung von Anwendungsfällen
- Prototyp für die Validierung von Anwendungsfällen

Vorgehen

- Skizze des Aufbaus der Benutzeroberfläche (auf Papier)
- Bekannte Beispiele und Umgebung (z.B. Konventionen des Betriebssystems) berücksichtigen
- Bedürfnisse der Anwender einbeziehen
- Nicht nur Optik, sondern auch Ablauf der Interaktionen berücksichtigen (z.B. „Objekt-Aktions-Prinzip“)
- Prototyp bauen

Als Ergebnis kann man erreichen

- Style Guide, Richtlinie zur Gestaltung der Optik und Interaktion
- Modell der Navigation in der Anwendung
- Variationspunkte festlegen (z.B. für Expertenmodus, spätere Erweiterungen u.ä.)
- Hilfesystem, Benutzerdokumentation

Beispiel für ein Konzept zum Aufbau der Oberfläche einer Webanwendung in [Abb. 4](#).

Zustandsdiagramm

Ein wichtiges Hilfsmittel für die Spezifikation der Navigation in einer Anwendung mit graphischer Benutzeroberfläche ist das Zustandsdiagramm.

Seine wichtigsten Elemente sind:

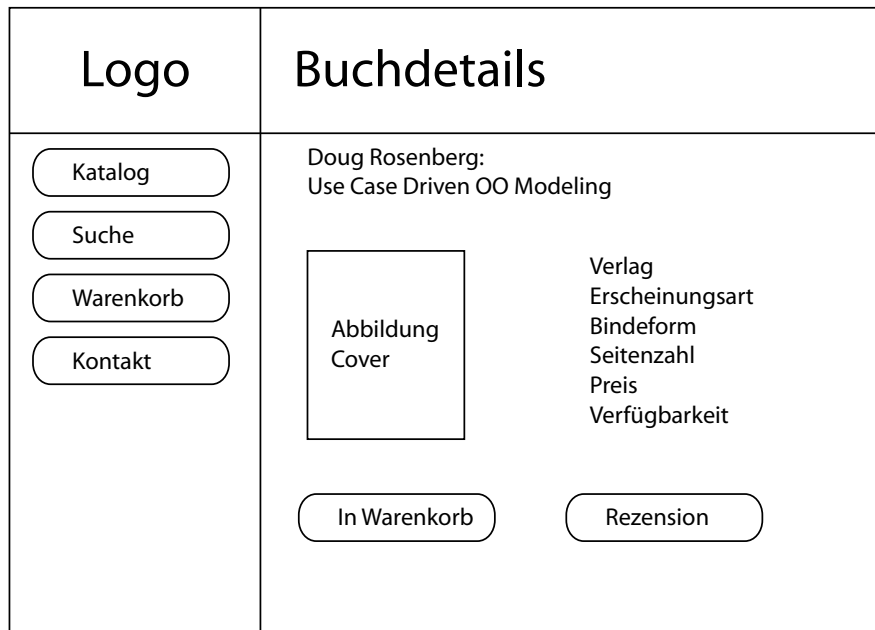


Abbildung 4: Konzept für eine Benutzeroberfläche

Zustand Eine über einen bestimmten Zeitraum andauernde Situation, die durch ein Ereignis verändert wird. Im Beispiel der Navigation entsprechen Zustände den Dialogen oder Masken des Programms, in denen der Benutzer Informationen angezeigt bekommt und Eingaben machen kann.

Ereignis Ein Ereignis verändert den Zustand. In unserem Fall der Navigation werden Ereignisse vom Anwender ausgelöst, in dem er auf einer Bildschirmseite seinen Button auslöst.

Aktion Eine Aktion ist an ein Ereignis geknüpft. Mit dem Ereignis wird eine Aktion ausgelöst, durchgeführt und anschließend wechselt das System in einen Folgezustand (der auch der bisherige Zustand sein kann).

Ein Beispiel, siehe Abb. 5 verdeutlicht die Verwendung des Zustandsdiagramms für die Festlegung des Navigationsmodells einer Anwendung

Szenarioanalyse

Die Szenarioanalyse (auch Stabilitätsanalyse oder „robustness analysis“) dient dazu, die bisherigen Schritte zu überprüfen und zusammenzubrin-

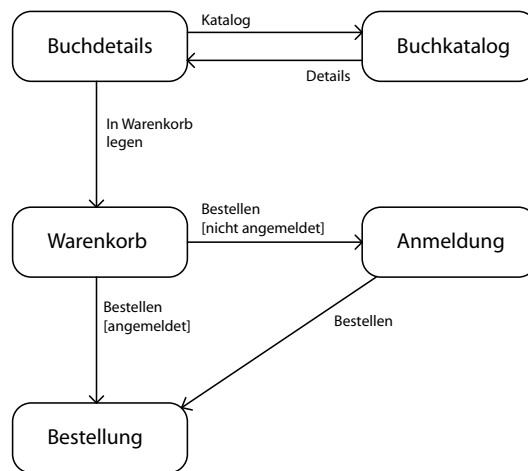


Abbildung 5: Navigation in einer Webanwendung – Zustandsdiagramm

gen.

Viele Anwendungen mit graphischer Benutzeroberfläche werden in einer sogenannten Schichtenarchitektur strukturiert. Die Schichten sind

- die Oberfläche für die Darstellung und die Interaktion mit dem Anwender (die *Präsentationsschicht*)
- die Komponenten, die für die Steuerung der Anwendung verantwortlich sind (die *Anwendungsschicht*)
- die Teile, die für das Speichern der verarbeiteten Daten, für ihre Persistenz zuständig sind (die *Persistenzschicht*)

In der Szenarioanalyse ordnet man nun die bisher über das *kS* gefundenen Sachverhalten diesen Schichten zu und überprüft damit die bisherigen Schritte der Anforderungsanalyse.

Die oben angegebene Architektur wird natürlich nicht für jedes Programm geeignet sein, sie kommt allerdings sehr häufig vor, insbesondere bei den Anwendungen im betrieblichen Bereich, bei den sogenannten Enterprise Anwendungen. Deshalb konzentrieren wir uns im Folgenden darauf.

Ziel

- Struktur des Systems finden: welche Komponenten sind in welcher Schicht erforderlich?

- Beziehungen der Komponenten untersuchen
- Steuerung der Abläufe im System überlegen
- bisherige Analyse: Fachmodell, Benutzeroberfläche und Anwendungsfälle validieren

Vorgehen

Die einzelnen Komponenten werden gemäß ihrem Charakter und ihrer Rolle den drei genannten Schichten zugeordnet. In der UML werden dazu Stereotype für die Klassen in diesen Schichten verwendet. Diese Stereotype haben jeweils ein eigenes Symbol für die Klassen:

- Klassen in der Präsentationsschicht haben die Stereotype <<boundary>>
- Klassen in der Anwendungsschicht haben die Stereotype <<control>>
- Klassen der Persistenzschicht haben die Stereotype <<entity>>

Diese Typen von Klassen werden visualisiert wie in Abb. 6 dargestellt.

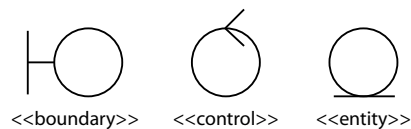


Abbildung 6: Typen von Klassen in der Szenarioanalyse

Die Zuordnung können wir nun aus unseren bisherigen Schritten herleiten:

- Die Elemente der Benutzeroberfläche, namentlich die Bildschirmseiten und die Dialoge sind Kandidaten für die Sorte <<boundary>>
- Die Klassen, die wir in der Domänenanalyse gefunden haben und ins Fachmodell übernommen haben, repräsentieren aufgrund unseres datenorientierten Vorgehens die zu speichernden Daten des Systems, also sind sie in der Regel Entity-Klassen.
- Schließlich verwenden wir die Anwendungsfälle und spielen sie durch. Aus diesem Vorgehen erkennen wir die benötigten Control-Klassen

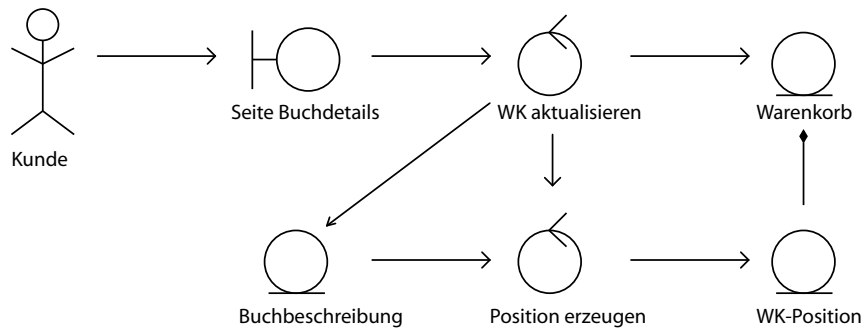


Abbildung 7: Beispiel Szenarioanalyse

Diagramme der Szenarioanalyse

Beispiel für ein Diagramm der Szenarioanalyse in Abb. 7.

Dabei bedeuten die Pfeile, dass die beteiligten Klassen zusammenarbeiten, um einen bestimmten Schritt zu erledigen. Sie bedeuten: „verwendet“, „steuert“. In dem Diagramm geht es noch nicht darum, konkrete Methoden zu spezifizieren, sondern darum zu sehen, ob die Bestandteile der Oberfläche mit den Elementen der Persistenzschicht so kommunizieren können, dass die Anwendungsfälle durchgeführt werden können. Und: welche steuernden Aktionen, verkörpert in Control-Klassen, benötigt werden um die Anwendungsfälle abzuarbeiten.

Beim Erstellen der Szenariodiagramme – in der Regel eines pro Anwendungsfall – wird geprüft, ob Fachmodell und Benutzeroberfläche so vorgesehen sind, dass das *kS* seinen Anforderungen gerecht werden kann.

Im Ergebnis der Szenarioanalyse verfügen wir über eine Spezifikation des *kS* in implementationsunabhängiger Form, eine sogenannte essentielle Lösung. Man spricht auch vom Analysemodell. Um mit ihm stehen wir auch schon im Entwurf – siehe nächste Vorlesung.

Lektüre: Udo Kelter Anwendungsfälle und Anwendungsfalldiagramme

http://pi.informatik.uni-siegen.de/kelter/lehre/04w/lm/lm_afd_info.html

Ein Beispiel, wie eine Spezifikation eines zu konstruierenden Systems im Ergebnis aussieht, sehen Sie an der Spezifikation des Musterprojekts für das SWT-Praktikum. Sie können die Dokumente zum Musterprojekt aus dem Kurs SWT im Moodle herunterladen.

Rev 2.1 – 16. Mai 2012