

## Use Cases considered harmful

Die Darstellung von Anwendungsfällen (Use Cases) ist ein Bestandteil der UML, die als Quasi-Standard für objektorientierte Beschreibungssprachen weit verbreitet ist. Anwendungsfall-Diagramme erlauben auszudrücken, welche Nutzer ein Informationssystem zu welchem Zweck einsetzen. Die Erstellung solcher Diagramme gilt als Vorstufe zu der Modellierung des Informationssystems selbst, bei der Klassen-, Zustands- und eine Reihe weiterer Diagramme eingesetzt werden.

Auf den ersten Blick wirkt es einleuchtend, darzustellen, wer ein System zu welchem Zweck nutzt. Deshalb und auch wegen ihrer Anschaulichkeit im Vergleich zu anderen Diagrammtypen sind Anwendungsfalldiagramme bei Entwicklern und Managern beliebt. Ihr Einsatz birgt jedoch Gefahren:

- \* Anwendungsfall-Diagramme sind nicht mächtig genug, um Geschäftsprozesse darzustellen; umgekehrt enthalten Geschäftsprozeßmodelle alle jene Aussagen, die in Anwendungsfall-Diagrammen untergebracht werden können. Wer also eine Geschäftsprozeß-Analyse macht, braucht keine Anwendungsfalldiagramme mehr. Wer umgekehrt Anwendungsfall-Diagramme einsetzt, macht das oft in der Erwartung, sich damit die – schwierigere, aber lohnende – Geschäftsprozeß-Analyse zu ersparen.
- \* Anwendungsfall-Diagramme sind kein objektorientiertes Beschreibungsmittel. Die Prinzipien der Datenabstraktion sowie der Generalisierung und Spezialisierung, die zu redundanzfreien Modellen führen, spielen bei ihnen keine Rolle. Aus diesem Grund verleiten sie dazu, Aussagen über das System in einer gefälligen, jedoch im späteren Entwicklungsprozeß schwer handzuhabenden Form festzuhalten.

Diese Einwände werden im folgenden näher ausgeführt.

### 1. Anwendungsfälle und Geschäftsprozesse

Beschreibungssprachen für Geschäftsprozesse erlauben (zumindest) folgende Sachverhalte auszudrücken: In welche Arbeitsschritte gliedert sich ein Prozeß? Wer führt die Arbeitsschritte durch? Welche Arbeitsschritte sind systemgestützt?. Wie hängen die Arbeitsschritte zeitlich und logisch miteinander zusammen, d.h. wo werden sie hintereinander bzw. kollateral ausgeführt, wo gabeln sich Arbeitsabläufe, wo synchronisieren sie sich?

Anwendungsfall-Diagramme erlauben einen Teil dieser Sachverhalte auszudrücken, allerdings nicht den zeitlichen und logischen Zusammenhang der Arbeitsschritte. An dessen Stelle treten zwei Beziehungen (erweitert und enthält), die manchmal fälschlich als Prozeßablauf interpretiert werden, nach der UML-Definition dies jedoch nicht sind. Kurz: *Anwendungsfälle beschreiben einen Geschäftsprozeß minus dem Prozeß*, also seinem Ablaufaspekt.

Aus dieser Beobachtung resultieren zwei ernsthafte Mängel der Anwendungsfall-Modellierung. Wie stellt man fest, ob eine Menge gefundener Anwendungsfälle – ein paar triviale findet man ja meist schnell – ausreicht, um das gesamte System zu beschreiben? Bei einer Geschäftsprozeß-Analyse hat man in Form des Ablaufs einen roten Faden: man muß sich mit Vor- und Nachbedingungen der Arbeitsschritte befassen; man muß überlegen, ob die Schritte, so wie sie aufeinanderfolgen, ein sinnvolles Ganzes ergeben; man hat ein Kommunikationsmittel mit den Anwendern, die ihre Abläufe schildern und in grafischer Form wiederfinden.

Copyright (c) 2001 Thomas Matzner, München

Man erkennt gleichartige Abläufe und kann sie zu einem einzigen Ablauf zusammenfassen. Man erkennt die Gesamtheit aller Schritte, die bis zum Erreichen eines geschäftlichen Ziels gegangen werden müssen.

Bei Anwendungsfall-Diagrammen fehlt der Ablauf-Aspekt. Zwar werden solche Diagramme in der Praxis manchmal entsprechend eines Prozeßablaufs gegliedert (z.B. indem die Anwendungsfälle von oben nach unten in ihrer annähernden zeitlichen Reihenfolge aufgezeichnet werden), jedoch enthält die Beschreibungssprache kein Mittel, um diese Aussagen klar und präzise festzuhalten. Einem Anwendungsfall-Diagramm ist deshalb viel schwieriger als einem Prozeßdiagramm anzusehen, ob es alle vorkommenden Fälle abdeckt. Das Fehlen der Ablaufstruktur läßt auch nicht erkennen, wo gleichartige Ablaufstrukturen vorliegen. Dies führt manche Modellierer dazu, viel zu viele Anwendungsfälle zu definieren, die sich inhaltlich vielleicht nur in einem Attributwert oder einem funktionalen Detail unterscheiden.

Der zweite Nachteil zeigt sich bei der Realisierung eines derart modellierten Systems. Informationssysteme müssen auf irgendeine Weise von den Geschäftsprozessen Notiz nehmen, die sie unterstützen. Sei es, daß sie eine Workflow-Komponente enthalten, die aktiv den Prozeß steuert, sei es, daß auf konventionelle Weise der Prozeß in Form von Prüfungen erlaubter Vorgänge bzw. Vorschlägen solcher Vorgänge implementiert ist. Zu einem bestimmten Zeitpunkt muß also der Geschäftsprozeß, soweit er auf das System durchschlägt, ohnehin betrachtet werden – warum ihn also in der Analysephase ausklammern?

Wer also nur ein Anwendungsfall-Modell ohne ein Geschäftsprozeß-Modell macht, versäumt etwas Wesentliches. Wer jedoch seine Geschäftsprozesse modelliert hat, könnte zwar hieraus durch Weglassen der Ablaufmerkmale leicht ein Anwendungsfall-Diagramm ableiten. Warum aber zusätzlich ein reduziertes Modell herstellen, wenn das vollständige, viel aussagekräftigere schon vorliegt?

## **2. Anwendungsfälle und objektorientierte Beschreibungen**

Ein wesentliches Qualitätsmerkmal von Modellen ist ihre Redundanzfreiheit. Die Klassenmodellierung als das Kernstück der objektorientierten Modellierung fördert dieses Ziel. Das Datenabstraktionsprinzip führt für jede Beschreibung von Informationen an exakt eine Stelle, die zugeordnete Klasse, an der diese Beschreibung zu plazieren ist. Das Generalisierungsprinzip ermöglicht es weitergehend, gleichartige Beschreibungen in unterschiedlichen Klassen in einer generellen Klasse zusammenzufassen.

Anwendungsfallmodelle verführen zur Redundanz. Ein Anwendungsfall ist ja nichts anderes als ein systemgestützter Arbeitsschritt, meist also ein Dialog zwischen System und Benutzer. Man fühlt sich an die Zeit vor der Einführung von Analysemethoden erinnert, als eine fachliche Systembeschreibung oft entlang der Dialoge strukturiert war. Beim Dialog wurden auch die in ihm abzuhandelnden Geschäftsregeln beschrieben: Die Kundennummer wird eingegeben, auf korrekte Prüfziffer geprüft, der Kunde angezeigt, sein Gesamtumsatz errechnet usw.

Der Modellierer steht jetzt bei der Beschreibung eines Anwendungsfalls vor einem nahezu unlösbaren Dilemma. Wenn er den Anwendungsfall einigermaßen detailliert beschreiben soll – und das wird vielerorts gewünscht, sind doch die Anwendungsfall-Diagramme so schön anschaulich – dann kommt er kaum darum herum, Aussagen über Geschäftsregeln zu treffen, die eigentlich ins Klassenmodell gehören. Da dieselbe Geschäftsregel in unterschiedlichen Anwendungsfällen greifen kann, entstehen dadurch Redundanzen in-

nerhalb der Anwendungsfälle und später auch zum Klassenmodell. Wenn er dieser Versuchung widersteht, kann die Beschreibung des Anwendungsfalles nur mehr recht dürftig ausfallen – streng genommen, dürfte er nur *benannt* werden und mit einem Hinweis versehen: Details bitte im Klassenmodell nachschlagen. Dann wäre es mit der vielgerühmten Anschaulichkeit auch schon zu Ende.

An dieser Stelle soll auch einem weitverbreiteten Märchen widersprochen werden, die Anwendungsfall-Modellierung würde helfen, Klassen zu finden. Das Gestalten von Klassen als Abstraktion zusammengehöriger Attribute und Operationen ist ein eigenständiger Schritt, der tiefe Einblicke in die Zusammenhänge eines Fachgebiets erfordert und weder durch eine Anwendungsfall- noch durch eine Geschäftsprozeßmodellierung unterstützt wird. Was solche Modelle liefern, sind nicht Klassen, sondern bestenfalls einige relevanten fachlichen Begriffe.

Um die letzte Behauptung zu erhärten, führe man sich den Anspruch an ein Klassenmodell vor Augen, lang- lebig zu sein und vielerlei Organisationsformen zur Durchführung einer bestimmten fachlichen Aufgabe zu unterstützen. Die Klassen etwa innerhalb einer Versicherung (z.B. Kunden, Verträge, Tarife, Risiken, Leistungen) sollen davon unabhängig sein, ob die Produkte durch versicherungseigene Vertreter, durch Makler verkauft oder vom Kunden direkt im Internet geordert werden. Diese Fälle würden drei wesentlich unterschiedliche Anwendungsfall- oder Geschäftsprozeßmodelle ergeben.

Zusammenfassend läßt sich für die Modellierung die Empfehlung geben, auf zwei Grundpfeilern aufzubauen: einem Geschäftsprozeß- und einem Klassenmodell. Auch dieser Weg ist innerhalb der UML gangbar; es gibt vielerlei Vorschläge für die Geschäftsprozeßmodellierung z.B. auf der Basis von Aktivitätsdiagrammen. Auf die Anwendungsfall-Modellierung kann man ohne Schaden zu nehmen verzichten.