

Anwendungsfall

Wechseln zu:[Navigation](#), [Suche](#)

Dieser Artikel erfüllt die [GlossarWiki-Qualitätsanforderungen](#) nur teilweise:

Korrektheit: 3 (zu größeren Teilen überprüft)	Umfang: 2 (wichtige Fakten fehlen)	Quellenangaben : 3 (wichtige Quellen vorhanden)	Quellenarten: 4 (sehr gut)	Konformität: 4 (sehr gut)
--	---	---	--------------------------------------	-------------------------------------

Ein Anwendungsfall (engl. Use Case) beschreibt, wie sich ein System oder eine Anwendung unter bestimmten Bedingungen verhält. Die Definition von Anwendungsfällen hat zum Ziel, zwischen den Beteiligten eines Projekts Einigung über das Verhalten und den Umfang eines Systems zu erhalten.

Die Anzahl von Anwendungsfällen ist nicht vorgeschrieben, es sollten so viele definiert werden, wie für die Anwendung notwendig sind.

Inhaltsverzeichnis

- 1 Definition (Spezifikation der „UML 2.5“, S. 637^[1] und „UML 2.5.1“, S. 639^[2])
 - 1.1 Übersetzung (Kowarschick)
- 2 Definition (Spezifikation der „UML 2.4.1“, formal/11-08-06, S. 597^[3])
 - 2.1 Übersetzung (Kowarschick)
- 3 Akteure
- 4 Beziehungen zwischen Use Cases
 - 4.1 Direkte Beziehungen
- 5 Use-Case-Spezifikation
- 6 Typische Fehler
- 7 Quellen
- 8 Siehe auch

1 Definition (Spezifikation der „UML 2.5“, S. 637^[1] und „UML 2.5.1“, S. 639^[2])

*UseCases are a means to capture the requirements of systems, i.e., what systems are supposed to do. The key concepts specified in this clause are **Actors**, **UseCases**, and subjects. Each UseCase's subject represents a system under consideration to which the UseCase applies. Users and any other systems that may interact with a subject are represented as Actors.*

A UseCase is a specification of behavior. An instance of a UseCase refers to an occurrence of the emergent behavior that conforms to the corresponding UseCase. Such instances are often described by Interactions.

1.1 Übersetzung (Kowarschick)

Use-Cases (**Anwendungsfälle**) sind Hilfsmittel, um die Anforderungen von Systemen zu erfassen, d. h., um festzulegen, was Systeme tun sollen. Schlüsselkonzepte, die in diesem Abschnitt spezifiziert werden, sind **Akteure**, **Use-Cases** und Subjekte. Jedes Subjekt eines Use-Cases repräsentiert ein System, auf das der Use-Case angewendet werden kann. Benutzer und andere Systeme, die mit einem Subjekt interagieren können, werden durch Akteure repräsentiert.

Ein Use-Case ist die Spezifikation eines Verhaltens. Eine Instanz eines Use-Cases verweist auf den Eintritt des zugehörigen Ereignisses. Derartige Instanzen werden häufig durch Interaktionen beschrieben.

2 Definition (Spezifikation der „UML 2.4.1“, formal/11-08-06, S. 597^[3])

Use cases are a means for specifying required usages of a system. Typically, they are used to capture the requirements of a system, that is, what a system is supposed to do. The key concepts associated with use cases are **actors**, **use cases**, and the **subject**. The subject is the system under consideration to which the use cases apply. The users and any other systems that may interact with the subject are represented as actors. Actors always model entities that are outside the system. The required behavior of the subject is specified by one or more use cases, which are defined according to the needs of actors.

...

Use cases, actors, and systems are described using use case diagrams.

2.1 Übersetzung (Kowarschick)

Use-Cases (**Anwendungsfälle**) sind Hilfsmittel, um benötigte Anwendungsmöglichkeiten eines Systems zu spezifizieren. Typischerweise werden sie verwendet, um die Anforderungen an ein System zu erfassen, d. h., zu erfassen, um festzulegen, was ein System tun sollen.

Die den Anwendungsfällen zugeordneten Schlüsselkonzepte sind **Akteure**, **Use-Cases** und das **Subjekt**. Das Subjekt ist das System, auf das die Use-Cases angewendet werden. Die Benutzer und beliebige andere Systeme, die mit einem Subjekt interagieren können, werden durch Akteure repräsentiert. Akteure modellieren stets Entitäten, die sich außerhalb des Systems befinden. Das erforderliche Verhalten des Subjekts wird durch ein oder mehrere Use-Cases spezifiziert, welche gemäß den Bedürfnissen von Akteuren definiert werden.

...

Use-Cases, Akteure und Systeme werden mit Hilfe von Use-Case-Diagrammen beschrieben.

3 Akteure

Es fällt auf, dass in der Version 2.5 im Gegensatz zur Version 2.4 nicht mehr gefordert wird, dass sich Akteure außerhalb des Systems befinden.

In der Spezifikation 2.4.1 wird noch explizit gefordert, dass sich ein Akteur außerhalb des Systems befinden muss:

Spezifikation der „UML 2.4.1“, formal/11-08-06, S. 598^[3]

An Actor models a type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data), but which is external to the subject (i.e., in the sense that an instance of an actor is not a part of the instance of its corresponding subject).

In der Spezifikation 2.5.1 wird die Rolle der Akteure wesentlich unspezifischer beschrieben:

Spezifikation der „UML 2.5.1“, S. 640^[2]

UseCases may have associated Actors, which describe how an instance of the Classifier realizing the UseCase and a user playing one of the roles of the Actor interact.

...

An Actor models a type of role played by an entity that interacts with the subjects of its associated UseCases (e.g., by exchanging signals and data). Actors may represent roles played by human users, external hardware, or other systems

Das heißt, in UML 2.5 ist es beispielsweise erlaubt, dass der Benutzer (human user) in der Rolle eines Avatars mit einem Use Case interagieren. Dies wird in diesem Wiki in Tutorien zur Spieleprogrammierung, wie beispielsweise [Pong](#), ausgenutzt: Der Benutzer selbst kann in einem Spiel meist nur einfache Aktionen ausführen (wie beispielsweise einen Schläger zu bewegen). Sein Stellvertreter im Spiel, d. h. sein Avatar (wie der Schläger in Pong) führt die eigentlichen spielentscheidenden Aktionen aus (Ball zurückschlagen, Ball abfälschen, Ball mit Punktverlust passieren lassen).

4 Beziehungen zwischen Use Cases

4.1 Direkte Beziehungen

Direkte Beziehungen zwischen Uses Cases kann es nur über Systemgrenzen hinweg geben:

Spezifikation der „UML 2.5.1“, S. 640^[2]

Two UseCases specifying the same subject cannot be associated as each of them individually describes a complete usage of the subject.

5 Use-Case-Spezifikation

Use-Case-Diagramme beschreiben das Verhalten eines Systems nur grob. Insbesondere modellieren sie keine zeitlichen Aspekte, d. h. keine Reihenfolge in der bestimmte Aktionen erfolgen können oder müssen. Daher ist es notwendig, im Anschluss an ein Use-Case-Diagramm die einzelnen Use-Cases genauer zu spezifizieren. Dazu werden häufig [Use-Case-Spezifikationen](#) eingesetzt (die, wenn sie auf speziellen Karten notiert werden, auch [Use Case Cards](#) heißen). Wenn man nach den beiden Begriffen googelt, stellt man schnell fest, dass es viele Vorschläge gibt, wie eine derartigen Spezifikation aussehen kann (siehe z. B. [Visual Paradigm: What is Use Case Specification?](#)).

Derzeit gibt es jedoch keine Norm dafür. Üblicherweise können/sollten folgende Informationen angegeben werden:

Name

beteiligte Akteure

Vorbedingungen, die erfüllt sein müssen, damit der Use Case ausgeführt werden kann

Nachbedingungen, die erfüllt sind, wenn der Use Cases erfolgreich ausgeführt wurde

Beziehungen zu anderen Use Cases

Kurzbeschreibung

detaillierte Beschreibung (evtl. mit weiteren UML-Diagrammen, wie anderen

[Anwendungsfalldiagrammen](#), [Aktivitätsdiagrammen](#), [Sequenzdiagrammen](#), [Zustandsdiagrammen](#) etc.)

Fehlerbehandlung

Priorität

Status

Autor

...

6 Typische Fehler

[Akteure und Aktionen](#) [Include und Extend](#) [Vererbung](#) [Systemgrenzen](#)



7 Quellen

OMG (UML 2.5): Object Management Group; OMG Unified Modeling Language - Version 2.5;

<http://www.omg.org/spec/UML/2.5>; 2015; Quellengüte: 5 (Web)

OMG (UML 2.5.1): Object Management Group; OMG Unified Modeling Language - Version 2.5.1;

<https://www.omg.org/spec/UML/2.5.1>; 2017; Quellengüte: 5 (Web)

OMG (UML 2.4.1): Object Management Group; OMG Unified Modeling Language - Version 2.4.1;

<http://www.omg.org/spec/UML/2.4.1>; 2011; Quellengüte: 5 (Web)

Rupp, Queins, SOPHISTen (2012): Chris Rupp, Stefan Queins und die SOPHISTen; UML 2 glasklar: Praxiswissen für die UML-Modellierung; Verlag: [Carl Hanser Verlag](#); ISBN: 3446430571, 978-3446430570; [Web-Link](#); 2012; Quellengüte: 5 (Buch)

Kowarschick (MMDB-Skript): Wolfgang Kowarschick; Vorlesung Multimedia-Datenbanksysteme - Sommersemester 2018; Hochschule: [Hochschule Augsburg](#); Adresse: [Augsburg](#); [Web-Link](#); 2018; Quellengüte: 4 (Skript)

Kowarschick (MMDB): Wolfgang Kowarschick; Vorlesung „Multimedia-Datenbanksysteme“; Hochschule: [Hochschule Augsburg](#); Adresse: [Augsburg](#); [Web-Link](#); 2016; Quellengüte: 3 (Vorlesung)

Kowarschick (MMProg): Wolfgang Kowarschick; Vorlesung „Multimedia-Programmierung“; Hochschule: [Hochschule Augsburg](#); Adresse: [Augsburg](#); [Web-Link](#); 2018; Quellengüte: 3 (Vorlesung) 8

Siehe auch

- [Unified Modeling Language](#)

Kategorien:

[UML](#)

[Glossar](#)

Diese Seite wurde zuletzt am 11. Juni 2019 um 15:20 Uhr bearbeitet.

Inhalt verfügbar unter [CC BY-SA 4.0](#).

