



[DOWNLOAD
DEMOVERSION](#)



[FEATURE
MATRIX](#)



[EA
KAUFEN](#)

Zustandsdiagramm (State Machine Diagram)

Zustandsdiagramme sind keine Erfindung der UML, sie gehen auf David Harel zurück, der sie entwickelt in den 80er Jahren, zurück. Diese Darstellungsform wurde in die UML

Ein Zustandsdiagramm zeigt eine Folge von Zuständen, die ein Objekt im Laufe der Zeit einnehmen kann, und die Ursachen der Zustandsänderungen. Man kann für ein Objekt die Änderung des Zustandes in Abhängigkeit von ausgeführten Operationen modellieren. Ein besonderer Wert wird auf die Übergänge von einem Zustand in den nächsten gelegt. Das Zustandsdiagramm zeigt, welche Operationen oder Ereignisse die Zustände des Objekts geändert werden können. Man ersieht, welche Belegungen die Attribute des Objekts vor dem Übergang besitzen müssen.

Ein Objekt kann als Zustandsdiagramm/-„System“ modelliert werden unter der Voraussetzung, dass eine Liste von Zuständen angegeben werden kann, für die gilt:

- ✓ Das Objekt befindet sich immer (zu jedem beliebigen Zeitpunkt seiner Existenz) in einem Zustand dieser Liste, anders ausgedrückt:
- ✓ Das Objekt befindet sich nie in keinem der genannten Zustände (wenn doch, dann in mindestens einem Zustand in der Liste).
- ✓ Nie in mehreren Zuständen Ihrer Liste zugleich (wenn doch, dann haben Sie die Zustände falsch gewählt).

Ein Objekt in einem Zustand kann dort ruhen, es ist aber auch möglich, in Zuständen vorzusehen.

Befindet sich ein Objekt in einem Zustand, dann können durchaus auch Subzustände modelliert werden, z. B. in einem untergeordneten Diagramm (Compositional Diagramm). Ist das Verhalten in einem Zustand prozeduraler Natur, dann kann d

natürlich auch ein Verhaltensdiagramm anderer Art sein.

Zustandsdiagramme müssen einen Anfangszustand und können einen Endzustand haben. Zustandsübergänge, sogenannte Transitionen, werden stets durch ein Ereignis (event) ausgelöst.



[DOWNLOAD
DEMOVERSION](#)



[FEATURE
MATRIX](#)



[EA
KAUFEN](#)

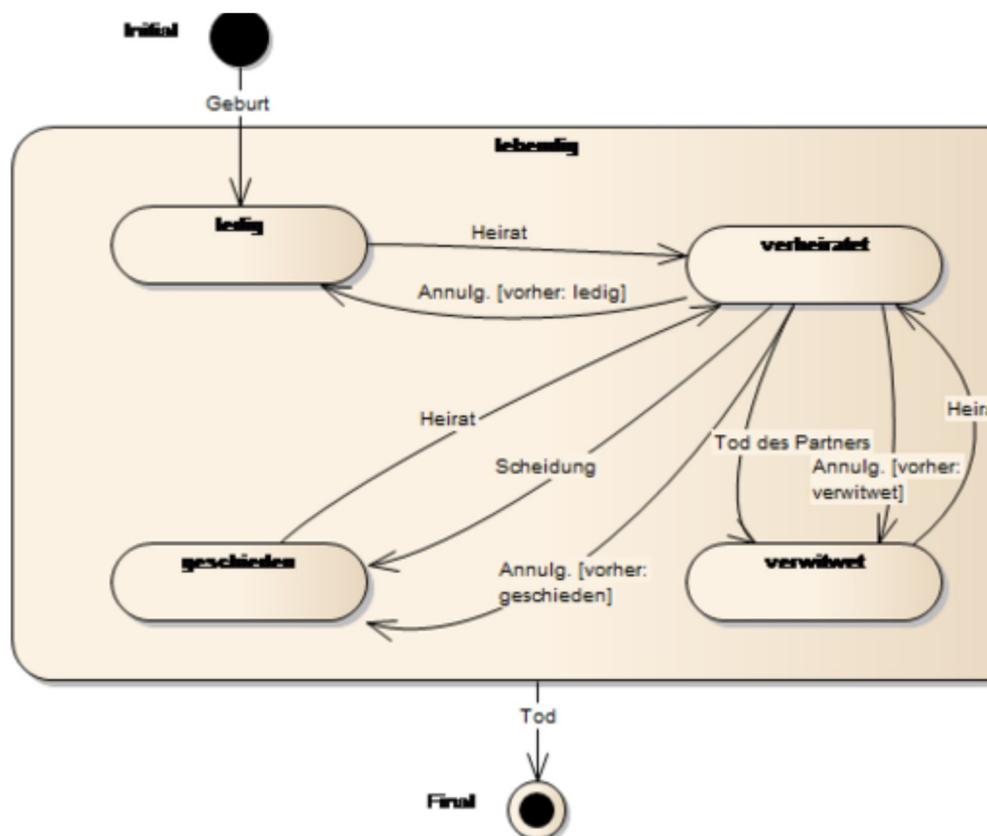


Abb. 28: Beispiel State-Machine-Diagramm

Zustände (States)

Zustände werden durch abgerundete Rechtecke modelliert. Sie können einen Namen, optional durch horizontale Linien in bis zu drei Bereiche eingeteilt werden. Im obersten Bereich steht der Name des Zustandes. Wenn der Name nicht angegeben wird, handelt es sich um einen Endzustand. In einem weiteren Bereich können existierende Zustandsvariablen mit ihren typischen Wertebelegungen angeführt werden. Der dritte Bereich innerhalb des Zustandes enthält eine Liste von internen Ereignissen, Bedingungen und aus ihnen resultierende Aktionen.

Ein Ereignis steht dabei für drei mögliche Verhaltensweisen:

- ✓ entry - löst automatisch beim Eintritt in einen Zustand aus, also bei allen here-Übergängen.

- ✓ exit - löst automatisch beim Verlassen eines Zustandes aus, also bei allen abg Übergängen.
- ✓ do - wird immer wieder ausgelöst, solange der Zustand nicht gewechselt wird



DOWNLOAD
DEMOVERSION



FEATURE
MATRIX



EA
KAUFEN

Übergänge von einem Zustand zum nächsten werden durch Ereignisse ausgelöst aus einem Namen und einer Liste möglicher Argumente. Ein Zustand kann Bedi Ereignis knüpfen, die erfüllt sein müssen, damit dieser Zustand durch dieses Erei wird. Diese Bedingungen können unabhängig von einem speziellen Ereignis sein

Gleichzeitig mit einem Zustandsübergang kann eine Aktion durchgeführt werden. Transition sieht folgendermaßen aus:

Ereignis [Bedingung] / Action

„Ereignis“, „[Bedingung]“ und „/Action“ sind allesamt optionale Bestandteile. Der Übergang vom Startpunkt zum ersten State darf fehlen. Auch an anderen Überg Ereignis weggelassen werden. Ist dies der Fall, wird der Zustand automatisch ge Aktivitäten des vorhergehenden Zustandes abgearbeitet wurden (entry..). Das KE wird auch als ANY-Trigger bezeichnet, dieses Ereignis ist IMMER vorhanden.

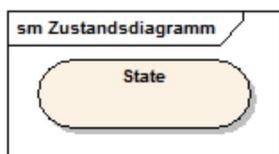
Symbole

Die folgende Tabelle enthält die Symbole der Zustandsdiagramme.

Name/Symbol

Verwendung

Zustand



Der Zustand eines Objekts wird durch ein Rechteck mit abg symbolisiert. Innerhalb dieses Symbols wird der Zustand be

Objekterzeugung



Der Startpunkt des Zustandsdiagramms wird mit einem ge dargestellt. Er ist identisch mit der Objekterzeugung. Nur ei State-Diagramm ist zulässig und muss vorhanden sein. Die Ar Startpunkts ist freigestellt.

Name/Symbol**Verwendung**

Objektzerstörung

Die Kette der Zustandsübergänge endet mit der Objektzerstörung.

**DOWNLOAD**
DEMOVERSION**FEATURE**
MATRIX**EA**
KAUFEN

der Activity im übergeordneten Diagramm, die das Unterdiagramm hat, zurück.

Übergang

Über einen Pfeil wird der Übergang (Transition) eingezeichnet. Dem Namen Auslösers (Trigger) beschriftet, die den Objektzustand können in eckigen Klammern eine Restriktion [Guard] angeben, nur bei deren Erfüllung der Objektzustand geändert wird. Zudem kann eine „/“ eine Aktivitätenliste, die beim Übergang auszuführen ist, angegeben werden. Guard und Aktivitätenliste sind optional, auch der Trigger (A) am Übergang von Initial und wenn ein ANY-Trigger modelliert.

Beispiel

Hochlauf eines Bankomaten und Hauptzustände. Beim Einschalten durchläuft der Bankomat einen Selbsttest. Abhängig davon, wie dieser ausgeht, wird die Grundstellung erreicht oder ein Störungszustand. Zusätzlich wurde vorgesehen, dass bei überlanger Dauer des Störungszustand eingenommen wird. Wird die Karte eingeschoben, erfolgt die Kartenzahlung. Abhängig von ihrem Ergebnis gelangt das Gerät in die Pin-Abfrage oder in den nächsten State. Weitere States wie Betragsabfrage, Verfügbarkeitsprüfung, usw. sind nicht mehr dargestellt.

Die Kettensymbole zeigen an, dass Subdiagramme vorhanden sind, die das Verhalten näher beschreiben. Es besteht die Freiheit als Subdiagramme beliebige Verhaltensweisen zu verwenden, es muss sich nicht unbedingt um weitere Zustandsdiagramme handeln.



[DOWNLOAD
DEMOVERSION](#)



[FEATURE
MATRIX](#)



[EA
KAUFEN](#)

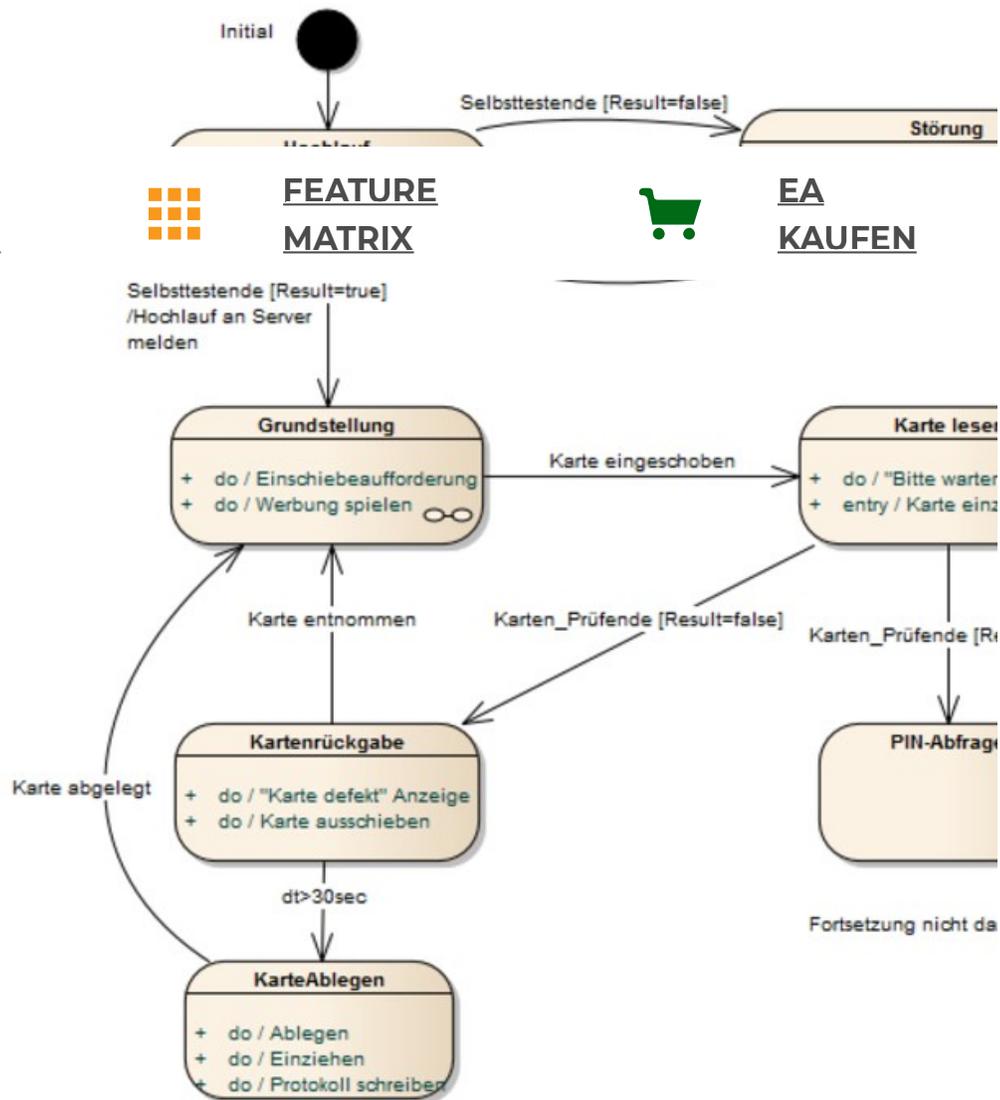


Abb. 29: Beispiel Zustandsdiagramm „Bankomathochlauf“



© 2019 Sparx Systems Ltd und SparxSystems Software GmbH.
Alle Rechte vorbehalten.

SparxSystems Software GmbH
Handelskai 340/Top 5 A-1020
Wien

Tel: +43 662 90 600 2041
Fax: +43 662 90 333 3041

Anfragen und Kommentare bitte an
✉ vendor@sparxsystems.eu



[DOWNLOAD
DEMOVERSION](#)



[FEATURE M](#)