

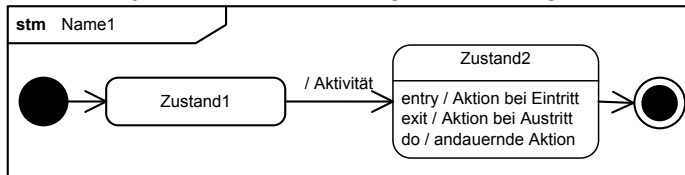
Die Symbolik der Unified Modelling Language (5)

Dynamische Modelle

Statische Modelle beschreiben den Aufbau und die Zusammenhänge zwischen den Komponenten eines Systems. Anhand eines Bauplans kann jedoch nicht das Verhalten eines Systems zur Laufzeit abgelesen werden. Für die Beschreibung von dynamischen Vorgängen bietet die UML eine Reihe von Diagrammen, die zur Planung und Veranschaulichung beim dynamischen Design ihren Einsatz finden.

Zustandsdiagramme

In der objektorientierten Programmierung können Objekte zur Laufzeit unterschiedliche Zustände annehmen. Zustandsdiagramme stellen diese Zustände und Zustandsübergänge eines Objekts dar.

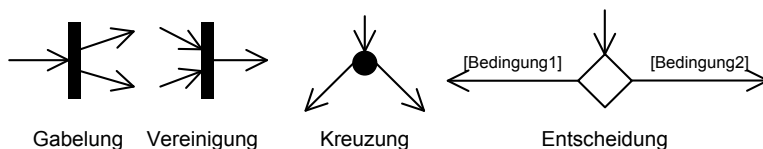
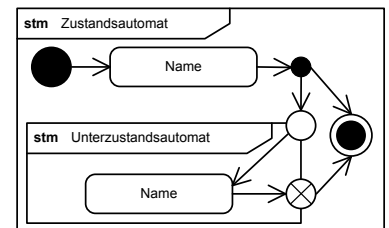
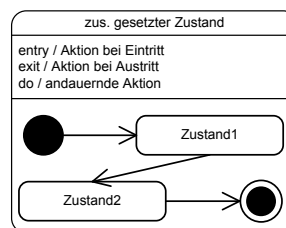


Das Symbol für einen Objektzustand ist ein abgerundetes Rechteck. Optional kann das Diagramm einen Rahmen mit der

Kennzeichnung „stm“ (statemachine) erhalten. Kreis und "Ochsenauge" symbolisieren dabei Anfangs- bzw. Endzustand des Diagramms. Zustandsübergänge (transitions) werden durch einen Pfeil dargestellt. Aktivitäten sind durch Schlüsselworte bezeichnet:

- entry: Aktionen beim Zustandseintritt,
- exit: Aktionen beim Verlassen des Zustands
- do: Aktionen während des betreffenden Zustands.

Mehrere Einzelzustände eines Systems können auch zu einem Gesamtzustand zusammengefasst werden. Da es sich bei der Beziehung wieder um eine Komposition handelt, bezeichnet man den Gesamtzustand auch als Komposit - Zustand (composit state). Ebenso ist es möglich, Zustandsautomaten ineinander zu verschachteln. Ein leerer Kreis bezeichnet hier den Eintritts-, der Kreis mit Kreuz den Austrittspunkt aus dem Unterzustandsautomat.



Über Gabelung und Vereinigung lassen sich parallele Abläufe und Synchronisation darstellen, eine Kreuzung bedeutet mehrere Möglichkeiten, wie es weitergehen kann.

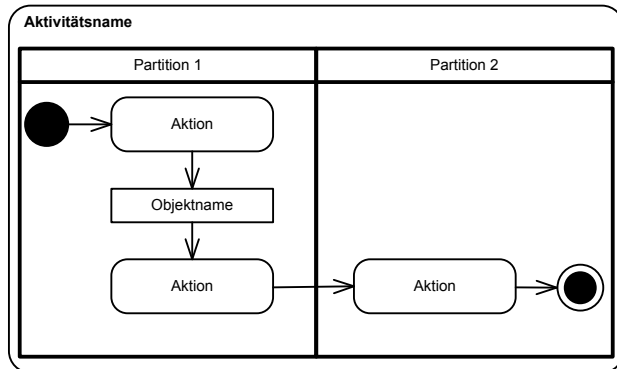
Entscheidungen werden mit Hilfe einer Raute symbolisiert. Die Bedingung (Wächter) steht in eckigen Klammern.

Als Sonderfall eines Zustandsautomaten gelten Protokollzustandsautomaten. Sie beschreiben alle möglichen Zustände und erlaubten Operationen eines Objekts einer Klasse abhängig von seinen Zuständen.

Die Symbolik der Unified Modelling Language (5)

Aktivitätsdiagramme

Aktivitätsdiagramme beschreiben die Aktivitäten eines Systems zur Laufzeit in Abhängigkeit von Zuständen. Sie bieten eine vereinfachte Sichtweise einer Operation oder eines Prozesses. Aktionen sind einzelne Schritte im gesamten Verarbeitungsablauf. Aktionen können dazu führen, dass sich der Zustand einzelner Objekte ändert.

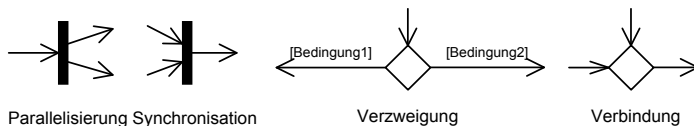


Der Übergang von einer Aktion zur nächsten wird durch einen Pfeil dargestellt (Kante).

Beim Aktivitätsdiagramm gilt das sogenannte „Tokenprinzip“. Es kann immer nur ein Pfad mit einem Token durchlaufen werden.

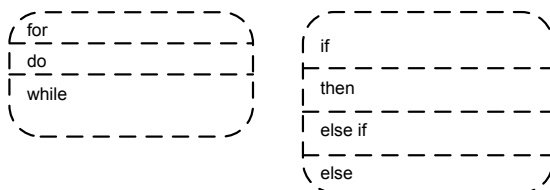
Partitionen („Schwimmbahnen“) können zur Darstellung der Verantwortlichkeit eingesetzt werden. Sie stellen dar, wer für welche Aktion verantwortlich ist.

Ein Splitting einer Aktion (Parallelisierung) wird durch eine senkrechte Linie und einer zulaufenden Kante, sowie mehreren abgehenden Kanten gekennzeichnet. Hier teilt sich das Token und mehrere



Aktionen können parallel nebeneinander ausgeführt werden. Die Synchronisation von Aktionen kennzeichnet man durch eine senkrechte Linie und mehreren zulaufenden Kanten sowie einer abgehenden Kante.

Bei Verzweigungen wird eine Kante an eine nicht ausgefüllte Raute geführt, die zu den weiteren Aktionen weiterleitet. Eine Verbindung führt mehrere Kanten zusammen.



Zur Programmsteuerung stehen Bereiche zur Verfügung. Damit lassen sich Auswahlentscheidungen und Schleifen modellieren. Die Aktionen werden dazu in die entsprechenden Bereiche gezeichnet.



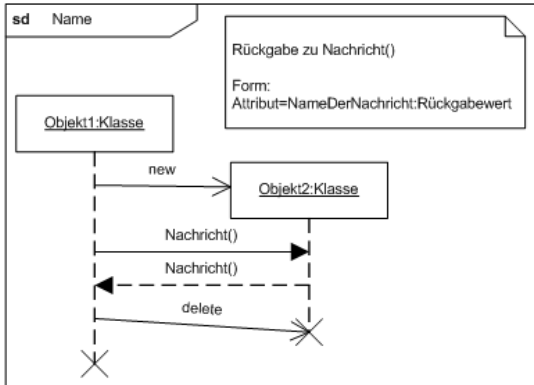
Ebenso können Signale gesendet oder Ereignisse empfangen werden. Sender und Ereignisempfänger stellen dabei eine Sonderform einer Aktion dar. Sobald das Signal gesendet bzw. das Ereignis empfangen ist, wird der Kontrollfluss weiter abgearbeitet.

Die Symbolik der Unified Modelling Language (5)

Interaktionsmodelle

Steht die Zusammenarbeit der Objekte im system im Vordergrund, so spricht man von so genannten Interaktionsmodellen. Sie erhalten einen Rahmen mit der Kennzeichnung „sd“ (Sequenzdiagramm).

Sequenzdiagramme



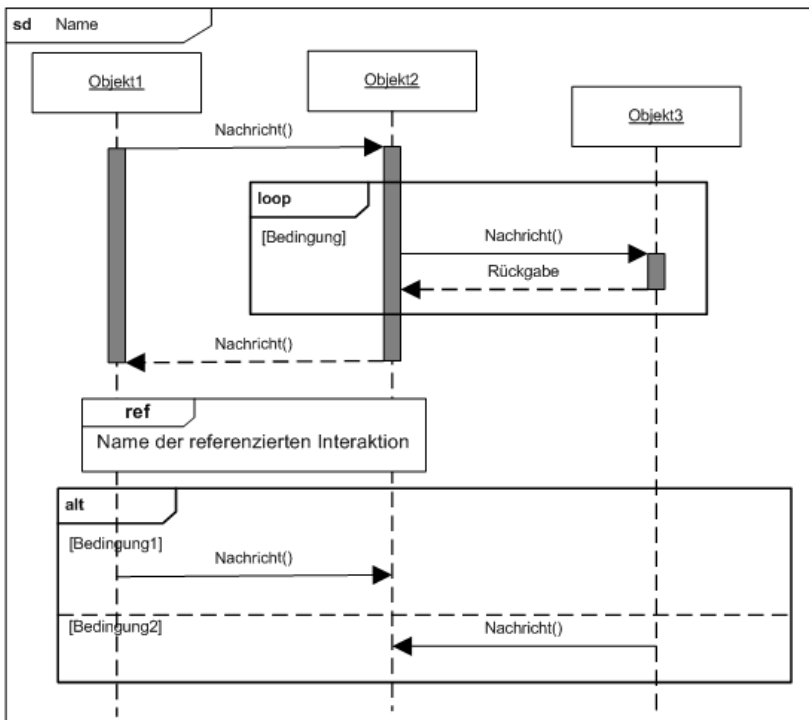
Beim Sequenzdiagramm wird der zeitliche Verlauf der Kommunikation zwischen den Objekten hervorgehoben. Die Zeitachse verläuft von oben nach unten, die Kommunikationspartner sind waagrecht eingezeichnet. Den Rahmen für Interaktionen kennzeichnet man mit sd (Sequenzdiagramm).

Objekte werden durch gestrichelte senkrechte Linien dargestellt (Lebenslinie). Über der Linie steht der Objektname. Ein X markiert das Ende der Objektlebenszeit im Programm. Die Nachrichten stellen waagerechte Linien zwischen den Objektlinien dar. Rückgaben einer Nachricht sind gestrichelt.

Das Sequenzdiagramm unterscheidet zwei Arten von Nachrichten: synchron und asynchron.

- **synchron** (geschlossene, ausgefüllte Pfeilspitze): Das sendende Objekt wartet auf eine Antwort.
- **asynchron** (offene Pfeilspitze): Die Nachricht wird ohne erwartete Rückgabe verschickt. Das sendende Objekt behält den Steuerfokus.

Breite Balken auf der Lebenslinie verdeutlichen eine Ausführungssequenz. Sie zeigen an, welches Objekt als Steuerungseinheit in diesem Zeitausschnitt anzusehen ist.



Die UML stellt für Sequenzdiagramme so genannte **Fragmente** zur Verfügung, mit deren Hilfe sich der Programmablauf beeinflussen lässt. Diese Rahmen sind mit vordefinierten Operatoren gekennzeichnet.

Für Schleifen in Sequenzdiagrammen stehen z.B. Fragmente mit der Bezeichnung **loop** zur Verfügung. Die Bedingung steht in Eckigen Klammern.

Auswahlentscheidungen im Ablauf kennzeichnet man in einem Fragment mit der Bezeichnung **alt**.

Einige weitere Interaktionsoperatoren sind z.B. **par** (Parallelverarbeitung), **opt** (optional) oder **critical** für atomare Aktionen.

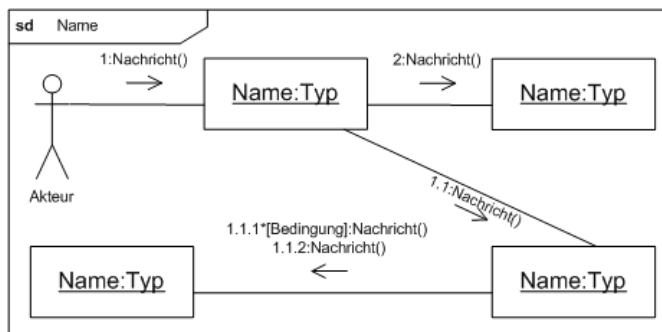
Damit komplexe Sequenzdiagramme übersichtlich bleiben ist es möglich, andere Interaktionen zu referenzieren. Die referenzierte Interaktion kann getrennt modelliert werden.

Alle Fragmente und Referenzen dürfen auch ineinander geschachtelt werden.

Die Symbolik der Unified Modelling Language (5)

Kommunikationsdiagramme

Beim Kommunikationsdiagramm stehen die Objekte und ihre Zusammenarbeit im Vordergrund. Kommunikationsdiagramme erhalten einen Rahmen mit der Kennzeichnung „sd“ (genau wie Sequenzdiagramme). Der zeitliche Verlauf der Nachrichten wird durch die Nummerierung der einzelnen

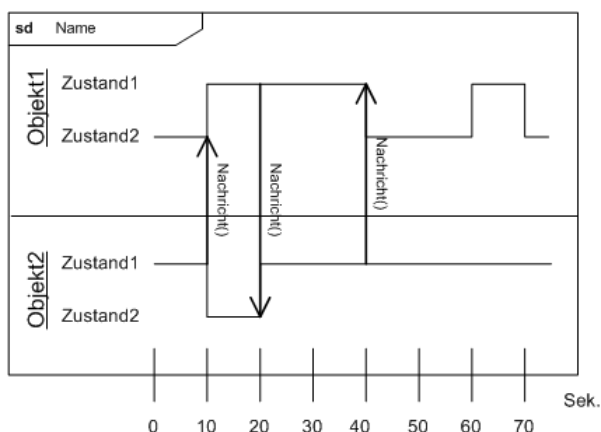


Nachrichten verdeutlicht. Unterpunkte (z.B. 1.1) beziehen sich auf die Schachtelungstiefe der Methodenaufrufe (z.B. in 1 wird MethodeA() aufgerufen, MethodeA() ruft mit 1.1 MethodeB() auf usw.). Es wird dargestellt, welche Nachrichten aus anderen Kommunikationsprozessen bereits abgeleistet sein müssen, bevor die anstehende Nachricht übermittelt wird. Nachrichten können dabei nicht nur zwischen Objekten sondern auch mit Aktionen, Akteuren, Interaktionsreferenzen

(siehe Interaktionsübersicht) usw. kombiniert werden. Die von einer Nachricht gelieferte Antwort kann zum besseren Verständnis mit einem Namen versehen werden. Die Richtung der jeweiligen Nachrichten zwischen den Objekten wird mit einem Pfeil markiert.

Eine Schleife (while) im Programmablauf kennzeichnet man bei Diagrammen durch ein Sternchen vor der Klammer (*[Schleifenbedingung]).

Timingdiagramme

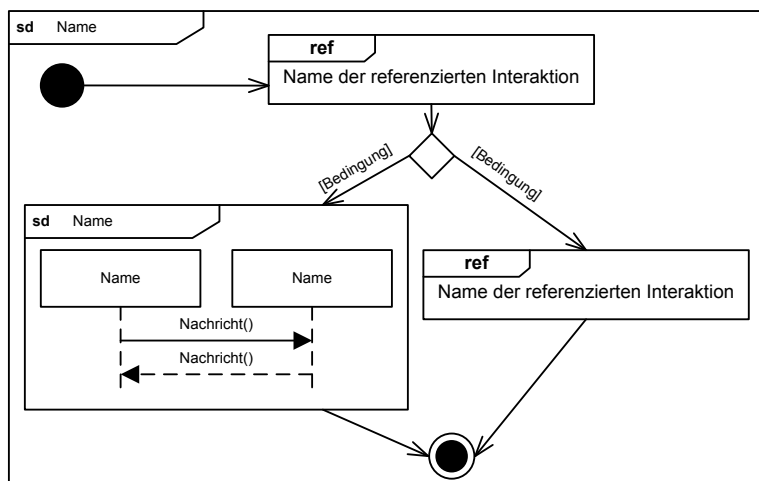


Timingdiagramme zeigen die Zustandsänderungen eines oder mehrerer Objekte im Kontext zum zeitlichen Ablauf. Der Schwerpunkt liegt dabei auf der Darstellung der Zeitskala. Die Zeitskala wird unten im Diagramm eingetragen.

Der Rahmen ist mit sd zu kennzeichnen, da es sich wieder um die Darstellung von Interaktionen handelt. Die waagrechten Linien spiegeln die Lebenslinie der Objekte in abhängigkeit von ihren Zuständen wieder. Nachrichten zwischen den Objekten symbolisieren senkrechte Pfeile.

Interaktionsübersichtsdiagramme

Kombiniert man mehrere Interaktionsdiagramme, so entsteht daraus eine Interaktionsübersicht. Sie beschreibt den Zusammenhang mehrerer Interaktionen.



Alle Interaktionsdiagramme (Rahmen mit sd) können für eine Interaktionsübersicht verwendet werden.

Ebenso ist es möglich, bereits detailgenauere Diagramme zu referenzieren. Damit bleibt die Übersicht gewahrt.

Für Start und Endpunkte, Verzweigungen und Verbindungen, Parallelisierung und Synchronisation werden die bereits vom Aktivitätsdiagramm bekannten Symbole verwendet.