

Interaktionsdiagramme

Udo Kelter

16.03.2003

Zusammenfassung dieses Lehrmoduls

Unter der Bezeichnung Interaktionsdiagramme faßt man zwei Diagrammtypen der UML (Sequenzdiagramme und Kollaborationsdiagramme) zusammen, durch die die Interaktion zwischen Objekten, also die Abfolge von Operationsaufrufen, spezifiziert und visualisiert werden kann. Beide Diagrammtypen unterscheiden sich inhaltlich nur wenig, Sequenzdiagramme betonen stärker die zeitliche Reihenfolge der Operationsaufrufe, während Kollaborationsdiagramme die Kommunikationsstruktur betonen. Dieses Lehrmodul stellt die grundlegenden Konzepte und Notationen der beiden Diagrammtypen vor; wir beschränken uns dabei auf den Fall sequentieller Systeme.

Vorausgesetzte Lehrmodule:

obligatorisch: – Objektorientierte Modellierung
empfohlen: – Anwendungsfälle und Anwendungsfalldiagramme

Stoffumfang in Vorlesungsdoppelstunden: 0.7

Inhaltsverzeichnis

1	Interaktionsdiagramme	3
1.1	Motivation und Einsatzgebiete	3
1.2	Arten von Interaktionsdiagrammen	3
2	Kollaborationen	4
3	Sequenzdiagramme	5
3.1	Lebenslinien	7
3.2	Operationsaufrufe	7
4	Kollaborationsdiagramme	9
5	Bemerkungen zum Entwicklungsprozeß	13
	Literatur	13
	Glossar	13
	Index	14

1 Interaktionsdiagramme

1.1 Motivation und Einsatzgebiete

Interaktionsdiagramme dienen dazu, Interaktionen zwischen Objekten zu spezifizieren bzw. zu illustrieren. Eine Interaktion ist dabei eine Folge von Operationsaufrufen innerhalb einer bestimmten Gruppierung von Objekten.

Eingesetzt werden Interaktionsdiagramme insb. dazu, Szenarios in Anwendungsfällen zu spezifizieren. Wie schon in Lehrmodul [AFD] erwähnt, beschreibt ein Anwendungsfall eine “Sequenz” von Arbeitsschritten.

Interaktionsdiagramme können aber nicht nur in der Analysephase, sondern auch später, insbesondere beim Entwerfen, eingesetzt werden, um Interaktionen zwischen beliebigen Objekten zu beschreiben. Motiviert ist ihr Einsatz dadurch, daß die statische Struktur, die durch Klassendiagramme modelliert wird, dynamische Abläufe oft nicht gut erkennen läßt. Ein Interaktionsdiagramm stellt dabei nur einen einzigen Ablauf dar; da i.a. sehr viele Abläufe denkbar sind, kann man aus Aufwandsgründen nur wenige besonders interessante Abläufe durch je ein Interaktionsdiagramm modellieren; ein Diagramm ist somit eher als ein Schnappschuß des Systemverhaltens und nicht als Teil einer vollständigen Spezifikation anzusehen.

Beim Einsatz in der Analyse werden üblicherweise weniger Details repräsentiert als beim Entwurf; manche syntaktische Konstrukte werden für die Analyse nur selten benutzt. Es gibt aber keine strikte Trennung zwischen Interaktionsdiagrammen für die Analyse bzw. für den Entwurf.

1.2 Arten von Interaktionsdiagrammen

Die UML definiert zwei Arten von Interaktionsdiagrammen: Sequenzdiagramme und Kollaborationsdiagramme.

Beide Diagrammtypen sind, wenn man vom Layout und einigen Details absieht, äquivalent. Ein Diagramm enthält nämlich in beiden Varianten

- mehrere Objekte
- die zwischen den Objekten verschickten Botschaften
- die Reihenfolge, in der die Botschaften verschickt werden.

Visualisiert werden dagegen unterschiedliche Aspekte:

- Ein **Sequenzdiagramm** (*sequence diagram*) zeigt vor allem den zeitlichen Ablauf durch eine Zeitachse an, durch die Interaktionen linear angeordnet werden.
- Ein **Kollaborationsdiagramm** (*collaboration diagram*) zeigt vor allem die Kommunikationsstruktur der beteiligten Objekte.

2 Kollaborationen

Interaktionsdiagramme beschreiben, wie schon erwähnt, Interaktionen zwischen mehreren Objekten. Diese Objekte werden nicht bunt zusammengewürfelt sein, sondern normalerweise irgendwie zusammengehören, also durch Beziehungen miteinander verbunden sein. Wenn wir beispielsweise wie in Bild 1 gezeigt eine Assoziation “betreut” zwischen Mitarbeitern und Kunden annehmen, kann sich eine Interaktion auf einen Mitarbeiter und einen Kunden beziehen, wobei der Mitarbeiter Betreuer des Kunden sein muß. Es könnte auch z.B. sein, daß ein Interaktionsdiagramm die Interaktion zwischen drei Komponent-Objekten eines Objekts beschreibt; das Gesamtobjekt erscheint dann nicht direkt als Sender oder Empfänger von Botschaften, ist aber natürlich relevant.

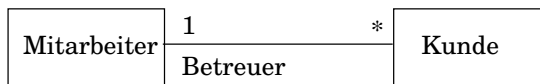


Abbildung 1: Beispiel für zwei durch eine Assoziation verbundene Klassen

Ein Interaktionsdiagramm bezieht sich daher immer auf eine “Gruppierung” von Objekten; eine solche Gruppierung wird in

[UML99] als **Kollaboration** bezeichnet. Die komplette Definition von Kollaborationen in der UML ist sehr komplex; wir geben sie hier nur vereinfacht wieder.

Objekte, die in Interaktionsdiagrammen auftreten, sind i.d.R. anonym, d.h. es wird nur der Klassenname angegeben. Dies drückt aus, daß die beschriebenen Interaktionen für beliebige Instanzen der Klassen gelten.

Tatsächlich ist allerdings weniger der Typ der Objekte entscheidend, sondern vielmehr der Sachverhalt, daß sie eine *Rolle* in einer bestimmten Beziehung spielen. Die Rolle impliziert den Typ, ist also die spezifischere Angabe. Umgekehrt impliziert der Typ nicht unbedingt die Rolle, in den meisten Situationen allerdings doch. Daher reicht es meist aus, die Kollaboration in Form eines Auszugs aus einem Klassendiagramm anzugeben, wobei dann maximal je ein Objekt der angegebenen Klassen in der Interaktion eine Rolle oder mehrere Rollen spielt. Andernfalls müssen zusätzlich die Rollen der beteiligten Objekte angegeben werden.

Ein gewisses Problem hat man in Analyse-Interaktionsdiagrammen mit Klassenoperationen. Bekanntlich werden Containerobjekte in der Analyse nicht explizit modelliert, sondern die Objektverwaltung wird implizit bei jeder Klasse unterstellt; ebenso werden Klassenoperationen der Basisklasse zugeordnet. Empfänger von Klassenoperationen ist aber die Klasse, kein einzelnes Objekt. Dies widerspricht der bisher grundlegenden Annahme bei Kollaborationen, daß anonyme Objekte (bzw. Rollen einer Beziehung) in eine Kollaboration involviert sind. In Analyse-Interaktionsdiagrammen werden daher bei Aufrufen von Klassenoperationen als Empfänger Klassen eingezeichnet.

Explizit dargestellt werden Kollaborationen i.d.R. nicht; sie ergeben sich implizit durch Klassen- und Interaktionsdiagramme.

3 Sequenzdiagramme

Die UML sieht Varianten von Sequenzdiagrammen für sequentielle und parallele Systeme vor; wir stellen hier nur die Variante für sequentielle

Systeme vor.

Ein Sequenzdiagramm benutzt zwei Dimensionen: Die Vertikale stellt eine Zeitachse dar, die Zeit schreitet von oben nach unten fort. Horizontal ist eine Sequenz von Objekten angeordnet, die in dem Szenario eine Rolle spielen¹. Eventuell vorhandene Beziehungen zwischen diesen Objekten werden nicht angezeigt.

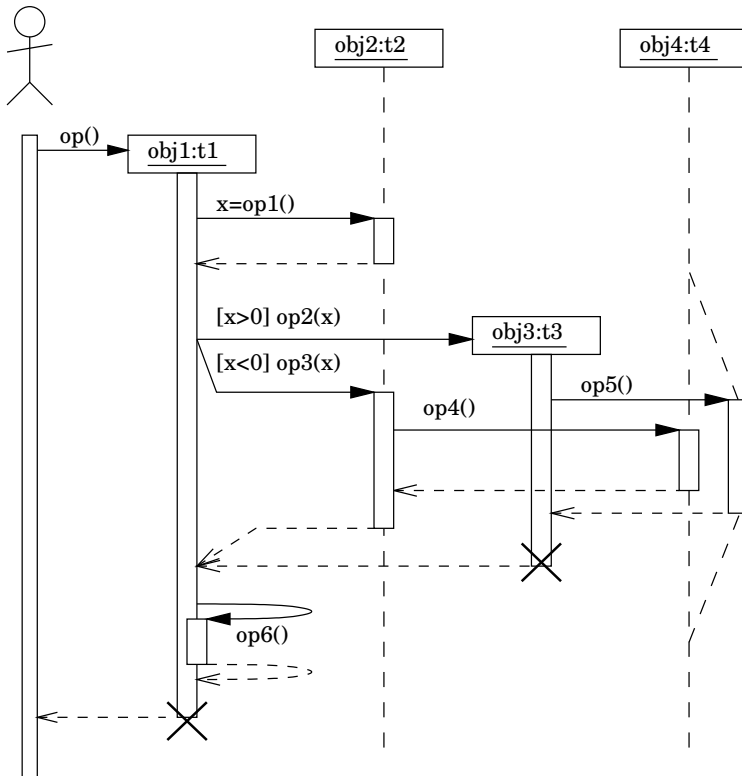


Abbildung 2: Beispiel eines Sequenzdiagramms

Die Gesamtfläche besteht daher i.w. aus einer Folge senkrecht stehender **“Lebenslinien”** von Objekten (s. Bild 2). Die Lebenslinien

¹Die UML erlaubt es auch, die beiden Dimensionen zu vertauschen.

werden gestrichelt gezeichnet.

Zwischen diesen Lebenslinien werden horizontal Operationsaufrufe, also Botschaften, eingetragen. Die Laufzeit der Botschaft wird hier als unwesentlich betrachtet, also nicht modelliert². Die Aufrufe werden als Pfeil mit durchgehender Linie gezeichnet und mit dem Namen der aufgerufenen Operation beschriftet. Der Rückkehrpfeil wird gestrichelt gezeichnet und bei sequentiellen Systemen oft einfach weggelassen; inhaltlich enthält er dann nämlich keine zusätzliche Information über das Szenario, weil keine Alternative besteht.

Die Anordnung der Objekte ist im Prinzip beliebig und hat keine inhaltliche Bedeutung. Man sollte die Objekte möglichst so anordnen, daß alle Aufrufpfeile in die gleiche Richtung (normalerweise nach rechts) gehen.

3.1 Lebenslinien

Wenn ein Objekt während der gesamten Dauer des Szenarios existiert, dann geht die Lebenslinie des Objekts von oben bis unten. Ein Objekt kann auch während des Szenarios erzeugt und/oder gelöscht werden; dann ist die Lebenslinie entsprechend lang. Der Löszeitpunkt wird zusätzlich durch ein großes X markiert.

Am oberen Ende der Lebenslinie wird ein Objektsymbol (vgl. [OOA]) für das betroffene Objekt eingetragen. Wenn das Objekt in dem Szenario erzeugt wird, muß es ein anderes Objekt geben, das eine erzeugende Operation aufruft. Der Aufrufpfeil geht dann seitlich in dieses Objektsymbol (z.B. Objekt `obj1` in Bild 2).

3.2 Operationsaufrufe

Der Rückgabewert eines Operationsaufrufs (z.B. von `op1` in Bild 2) kann einer Variablen zugewiesen und in späteren Operationsaufrufen verwendet werden.

²In einem verteilten System, in dem die Objekte auf verschiedenen Rechnern existieren, kann dies eine unzulässige Vereinfachung sein. Auf verteilte Systeme gehen wir hier nicht ein.

Ein Operationsaufruf kann bedingt sein; in Bild 2 ruft beispielsweise Objekt `obj1` im Fall `x > 0` die Operation `op2` bei Objekt `obj3` auf.

Bei bedingten Operationsaufrufen kann der Fall auftreten, daß bei *einem* Objekt *unterschiedliche* Operationen abhängig von den Bedingungen ausgeführt werden. In Bild 2 wird auf Objekt `obj4` entweder die Operation `op4` oder `op5` ausgeführt; welche, hängt von dem bedingten Aufruf in Objekt `obj1` ab. Die beiden alternativen Operationsausführungen werden parallel dargestellt. Die Lebenslinie von `obj4` wird hierzu vor diesen Operationsausführungen geteilt und anschließend zusammengeführt.

Die Bedingungen können sich auch auf die Parameter beziehen, mit denen die aufrufende Operation selbst aufgerufen worden ist. In diesem Fall müssen statt der leeren Parameterlisten die Namen der Parameter angegeben und damit sozusagen deklariert werden, z.B. `op3(y,z)`. Hier könnte dann ein folgender Operationsaufruf die Bedingung `[y < z]` haben.

Neben Bedingungen könnten auch Wiederholungen durch einen Stern vor dem Namen der aufgerufenen Operation dargestellt werden. In der Notation ** [bedingung] operation()* kann zusätzlich eine Abbruchbedingung für die Wiederholungen angegeben werden. Allerdings sind Interaktionsdiagramme nicht dazu gedacht und auch nicht dazu geeignet, komplexe Ablaufstrukturen zu spezifizieren; hierfür sind andere Diagrammtypen (insb. Zustandsübergangsdigramme und Aktivitätsdiagramme) eine bessere Wahl.

Wenn ein Diagramm durch Fallunterscheidungen bei Operationsaufrufen zu kompliziert wird, kann man stattdessen für jeden Fall ein eigenes Diagramm herstellen. Der Arbeitsaufwand erhöht sich dadurch natürlich. Man muß im Einzelfall abwägen, ob mehrere, weniger komplexe Diagramme die bessere Lösung sind.

Wiederholungen und Fallunterscheidungen wird man vor allem beim Entwurf einsetzen, weniger in der Analyse.

Eine Operation kann natürlich auch Operationen auf dem gleichen Objekt aufrufen. In diesem Fall führt der Aufrufpfeil wieder auf die

gleiche Lebenslinie zurück (s. Aufruf von `op6` in Bild 2).

Während des Zeitraums, in dem eine Operation auf einem Objekt ausgeführt wird, ist das Objekt “aktiviert”. Der Aktivierungszeitraum wird durch ein Rechteck visualisiert. Bei einem Operationsaufruf auf dem gleichen Objekt wird das Rechteck für die nächste Aktivierung etwas versetzt gezeichnet, um einen Stapel von Operationsausführungen zu visualisieren (s. Aufruf von `op6` in Bild 2).

Sequenzdiagramme können in unterschiedlichem Detaillierungsgrad benutzt werden. Bei der Analyse, wo sie vor allem zur Beschreibung der Abläufe innerhalb eines Anwendungsfalls eingesetzt werden, wird man i.a. nur grobe Strukturen angeben. Beim Entwurf wird man stattdessen einzelne Operationsaufrufe der beteiligten Objekte anführen.

4 Kollaborationsdiagramme

Ein Kollaborationsdiagramm ist ein gerichteter Graph. Knoten dieses Graphen sind die Objekte (bzw. Rollen), die in der Kollaboration, auf die sich das Kollaborationsdiagramm bezieht, auftreten. Kanten des Graphen sind die Beziehungen zwischen diesen Objekten. Annotiert werden die Beziehungen mit den darauf basierenden Operationsaufrufen. Bild 3 zeigt ein Beispiel, das inhaltlich dem Sequenzdiagramm in Bild 2 entspricht.

Die Kanten werden, sofern Analysemodelle vorliegen, ohne Pfeilspitze gezeichnet. Die Richtung des Operationsaufrufs wird durch einen separaten kleinen Pfeil angegeben.

In einem Kollaborationsdiagramm kann durch eine passende Anordnung der Objekte veranschaulicht werden, welche Objekte “eng” zusammenarbeiten.

Die zeitliche Anordnung der Operationsaufrufe, die ja bei Sequenzdiagrammen im Vordergrund stand, wird in Kollaborationsdiagrammen nicht graphisch ausgedrückt, sondern durch Nummern vor den Operationsaufrufen (s. Bild 3). Die von außen kommende Botschaft, die das Szenario auslöst, erhält keine Nummer. Alle Operationsaufrufe

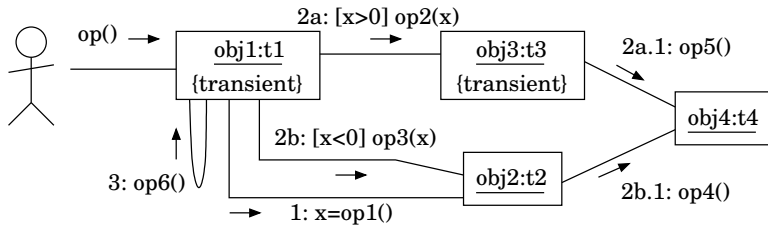


Abbildung 3: Beispiel eines Kollaborationsdiagramms

dieses Objekts werden beginnend mit 1 durchnummeriert. Bei Alternativen werden Zeichenfolgen verwendet (z.B. 1a, 1b). Sofern Operationen verschachtelt ausgeführt werden, wird dies durch entsprechende hierarchische Nummern ausgedrückt.

Analog zu Sequenzdiagrammen können Bedingungen, unter denen eine Operation aufgerufen wird, in eckigen Klammern angegeben werden.

Iterationen, also mehrfaches Senden einer Nachricht, können durch einen * angegeben werden; hierbei kann zusätzlich die Zahl der Durchläufe in eckigen Klammern angegeben werden, beispielsweise in der Form:

* [i := 1..n] op4()

Die zwischen den eckigen Klammern stehende Iterationsklausel kann in Pseudocode oder in einer konkreten Programmiersprache angegeben werden, die UML läßt das Format offen³. Dies gilt auch für die Bedingungen bei bedingten Operationsaufrufen.

Darstellung der Lebensdauer von Objekten. Aus der graphischen Darstellung eines Kollaborationsdiagramms kann man auch

³s. Abschnitt 3.72 in [UML99]. Jemand, der auf Basis einer solchen "Spezifikation" unterstützende Werkzeuge, die höhere Ansprüche an die Konsistenzüberprüfung von Dokumenten erfüllen, bauen soll, bekommt hier mittlere Alpträume. Faktisch müssen Werkzeughersteller solche Lücken in der Spezifikation durch private Entscheidungen füllen, was die Entstehung von Dialekten fördert und die Portabilität von Dokumenten zwischen unterschiedlichen Werkzeugen behindert.

nicht entnehmen, ob ein Objekt in dem dargestellten Szenario erzeugt und/oder gelöscht wird. Dies kann man aber durch Constraints ausdrücken; vordefiniert sind:

- `{new}` das Objekt wird erzeugt, aber nicht gelöscht
- `{destroyed}` das Objekt existierte schon vorher und wird gelöscht
- `{transient}` das Objekt wird erzeugt und gelöscht

Bezugszeitrahmen ist jeweils der Zeitraum, den die dargestellten Interaktionen einnehmen. Bild 3 zeigt als Beispiel zwei Objekte mit dem Constraint `{transient}`. Neben Objekten können auch Beziehungen in gleicher Weise gekennzeichnet werden.

Referenzierung von Objekten. Manchmal ist es bei einem Operationsaufruf zusätzlich interessant zu wissen, auf welche Art und Weise die aufrufende Operation Zugriff auf das benutzte Objekt erhalten hat. Hier definiert die UML folgende Stereotypen:

- `<<association>>` Das aufrufende Objekt hat eine Beziehung zu dem aufgerufenen; dies ist der Normalfall, deshalb kann dieses Stereotyp auch weggelassen werden, kann aber dennoch aufgeführt werden, wenn dieser Sachverhalt besonders betont werden soll.
- `<<global>>` Das aufgerufene Objekt ist global bekannt.
- `<<local>>` Das aufgerufene Objekt muß hier `{transient}` oder `{new}` sein; es wurde in der aufrufenden Operation erzeugt.
- `<<parameter>>` Das aufgerufene Objekt ist der aufrufenden Operation als Parameter übergeben worden.
- `<<self>>` Aufrufendes und aufgerufenes Objekt sind identisch.

Notiert werden diese Stereotypen neben der Verbindungslinie zwischen den beiden involvierten Objekten, und zwar nahe bei dem aufgerufenen Objekt.

Multi-Objekt-Symbol. Wenn man Kollaborationsdiagramme bei der Analyse einsetzt, können Operationen auftreten, die mit Kollektionen von Objekten arbeiten. Nehmen wir als Beispiel eine Operation an, die eine Liste von Namen der Mitarbeiter einer Abteilung anzeigt, wobei deren Anzahl zunächst im Kopf der Liste angezeigt wird. Die notwendige Abfrage der Anzahl richtet sich an die Gesamtmenge der Adreßobjekte.

Eine derartige Menge kann in Kollaborationsdiagrammen durch ein **Multi-Objekt-Symbol** dargestellt werden. Das Symbol besteht aus einem Symbol für ein anonymes Objekt, das nach oben rechts mit einem Schatten versehen ist, um einen Stapel anzudeuten.

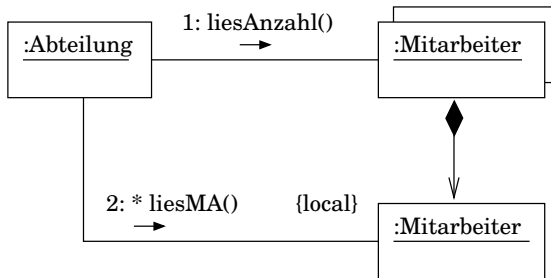


Abbildung 4: Beispiel für ein Multi-Objekt-Symbol

Ähnlich wie die Operation, die die Anzahl der Mitarbeiter liest, wäre eine Operation zu behandeln, die Referenzen auf alle Mitarbeiter oder eine irgendwie selektierte Teilmenge daraus liefert. Nach Vorliegen der Referenzen muß anschließend für jede Referenz eine Operation (`liesMA()`) für das jeweilige Objekt aufgerufen werden, das die hier interessierenden Mitarbeiterdaten liest. In Bild 4 wird dies durch den unteren, iterierten Operationsaufruf dargestellt. Man beachte, daß für das einzelne Mitarbeiter-Objekt ein separates Symbol benötigt wird, das Multi-Objekt-Symbol kann hier nicht benutzt werden.

Zwischen dem Multi-Objekt-Symbol und dem Symbol für ein einzelnes Objekt aus der Menge kann zusätzlich eine Kompositionsbeziehung eingetragen werden.

5 Bemerkungen zum Entwicklungsprozeß

Es wurde schon darauf hingewiesen, daß Sequenz- und Kollaborationsdiagramme inhaltlich i.w. äquivalent sind. Bevor man ein Sequenzdiagramm aufschreibt, sollte man sich allerdings über die exakte Sequenz im klaren sein; selbst bei einem guten Werkzeug werden nachträgliche Änderungen an der Folge der Operationsaufrufe Umstände bereiten. Bei Kollaborationsdiagrammen kann man zunächst völlig auf die Angabe der Reihenfolge der Aufrufe verzichten, wenn dies anfangs noch nicht völlig durchdacht ist.

Literatur

- [Ba99] Balzert, Heide: Lehrbuch der Objektmodellierung; Spektrum Akademischer Verlag; 1999
- [UML99] OMG Unified Modeling Language Specification (draft, Version 1.3 alpha R5, March 1999); OMG; 1999
- [AFD] Kelter, U.: Lehrmodul “Anwendungsfälle und Anwendungsfall-diagramme” 2003
- [OOA] Kelter, U.: Lehrmodul “Objektorientierte Modellierung”; 2003

Glossar

Interaktionsdiagramm (*interaction diagram*): stellt eine oder mehrere Interaktionen zwischen Objekten dar; eine Interaktion ist dabei eine Folge von Operationsaufrufen innerhalb einer bestimmten Gruppierung von Objekten; ist ein Sammelbegriff, unter dem Sequenzdiagramme und Kollaborationsdiagramme zusammengefaßt werden

Kollaborationsdiagramm (*collaboration diagram*): Diagramm, das Operationsaufrufe zwischen mehreren Objekten (bzw. innerhalb einer Kollaboration) darstellt, wobei die Kommunikationsstruktur besonders hervorgehoben wird

Kollaboration (*collaboration*): Gruppe von miteinander kommunizierenden, i.d.R. durch Beziehungen verbundenen, meist anonymen Objekten; in Ausnahmefällen enthält die Gruppe auch Klassen

Lebenslinie (*life line*): Linie in einem Sequenzdiagramm, die die Aktivitäten eines Objekts darstellt; die Lebenslinie ist als gestrichelte Linie gezeichnet, wenn das Objekt auf einen Operationsaufruf wartet, und als ein schmales Rechteck während jeder Operationsausführung

Sequenzdiagramm (*sequence diagram*): Diagramm, das die zeitliche Abfolge von Operationsaufrufen zwischen mehreren Objekten (bzw. innerhalb einer Kollaboration) darstellt, wobei der zeitliche Ablauf besonders hervorgehoben wird

Index

- Analysephase, 3
- association, 11
- Assoziation, 4, 9
- collaboration diagram*, 4
- destroyed, 11
- Entwicklungsprozeß, 13
- global, 11
- Interaktionsdiagramm, 3, 13
- Kollaboration, 4, 5, 13
 - Darstellung, 5
- Kollaborationsdiagramm, 4, 9, 13
- Lebenslinie, 6, 7, 14
- local, 11
- Multi-Objekt-Symbol, 12
- new, 11
- Objekt
 - aktives, 9
 - anonymes, 5
 - Beziehungen, 6
 - Erzeugung, 10
 - Klassenoperationen, 5
 - Rolle, 5
- Objektsymbol, 7
- Operationsaufruf
 - bedingter, 7
 - iterierter, 8, 10
 - Numerierung, 9
 - Stereotyp, 11
- parameter, 11
- self, 11
- sequence diagram*, 4
- Sequenzdiagramm, 4, 5, 14
- transient, 11