

Übersicht

1 Einführung

2 Aktivitätsdiagramme

2.1 Notationselemente

2.2 Entwicklung von UML 2

2.3 Anwendung

3 Zusammenfassung

4 Literatur



Einführung

Aktivitätsdiagramme

♦ Notationselemente

♦ Entwicklung

♦ Anwendung

Zusammenfassung

Literatur

Kontakt

1 Einführung :

Aktivitätsdiagramm ist ein wichtiges Diagramm für die Modellierung der Ablaufmöglichkeiten eines Systems .

Im Vortrag sind drei Fragen zu klären

- Frage 1: Wie modelliert man ein bestimmtes Systemverhalten ?
- Frage 2: Wozu braucht man Aktivitätsdiagramme?
- Frage 3: Welche Unterschiede gibt es zwischen UML 2 und UML 1.x ?

Was ist ein Aktivitätsdiagramm ?

Aktivitätsdiagramme stellen einen Diagrammtyp dar, der den Fluss von Aktivität zu Aktivität zeigt . Ein Aktivitätsdiagramm dient der Darstellung des dynamischen Systemverhalte .

2.1 Notationselemente

a) Aktion : Eine Aktion steht für die Beschreibung eines Verhaltens oder die Bearbeitung von Daten ein Aktion wird, die innerhalb einer Aktivität nicht weiter zerlegt wird .



Aktionname



laufen

Es gibt noch Sonderformen von Aktion.(SendSignalAction und AcceptEventAction)



Sende
Signal



Reparatur
ausgeführt

Einführung

Aktivitätsdiagramme

♦ Notationselemente

♦ Entwicklung

♦ Anwendung

Zusammenfassung

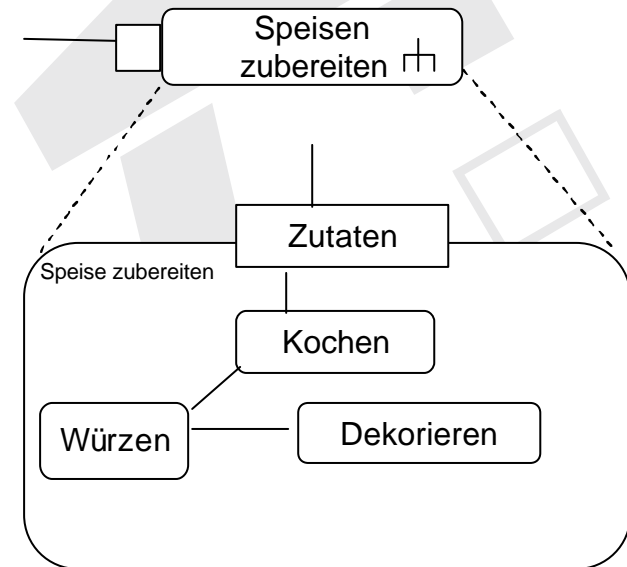
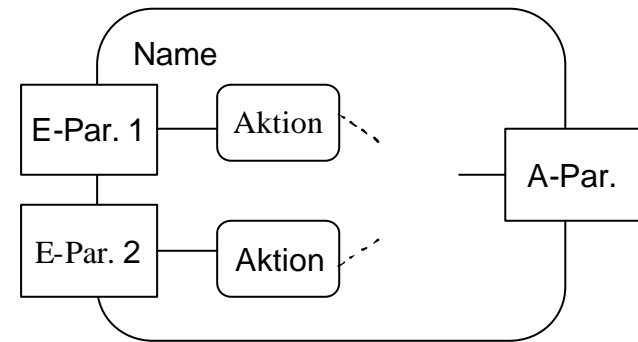
Literatur

Kontakt

b) Aktivität :

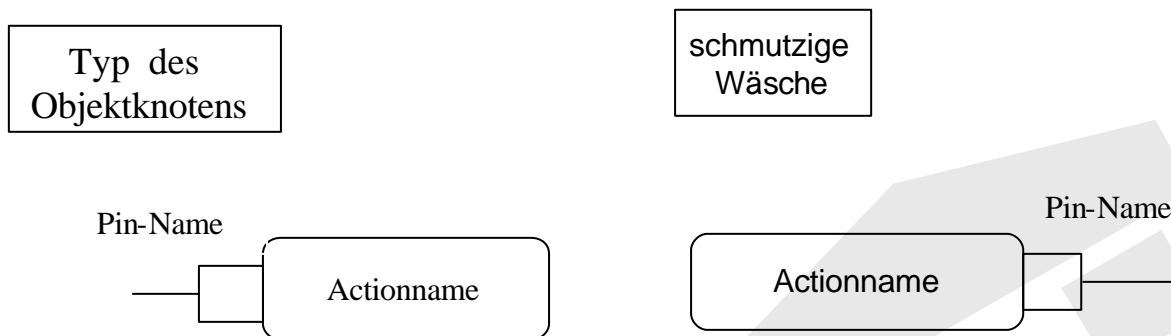
Eine Aktivität besteht aus einer Folge von Aktionen und weiteren Elementen .

Eine Aktivität kann geschachtelt werden , da eine Aktivität wieder eine Aktivität auslösen kann .

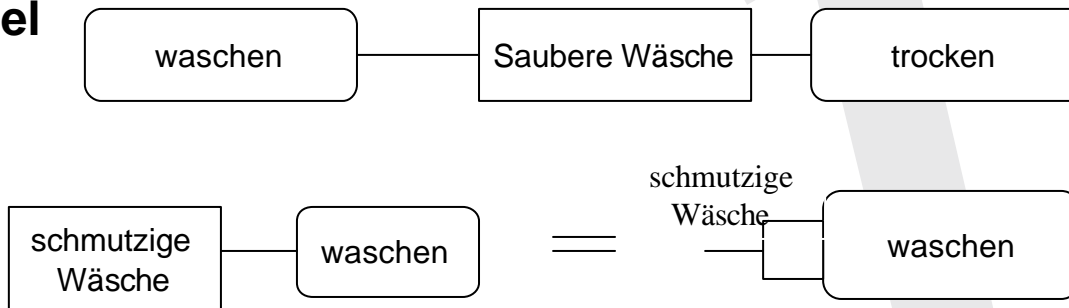


c) Objektknoten

Objektknoten transportieren Daten und Werte innerhalb einer Aktivität während eines Ablaufs. Der Pin-Name ist der Objektknoten, der mit der Aktion verbunden ist.



Beispiel
:



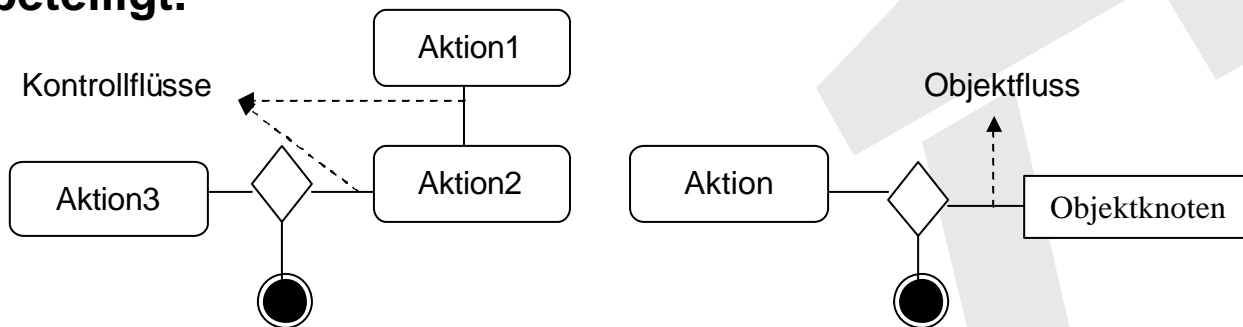
d) Kanten

Kanten sind Übergänge zwischen zwei Knoten .

Die Kanten werden in zwei Arten unterteilt :

Kontrollfluss : Es findet zwischen zwei Aktionen, oder einer Aktion und einem Kontrollelement, ein Fluss statt (d.h. die Abarbeitung wird mit der Kante vorgeführt).

Objektfluss : An dieser Kante ist mindestens ein Objektknoten beteiligt.

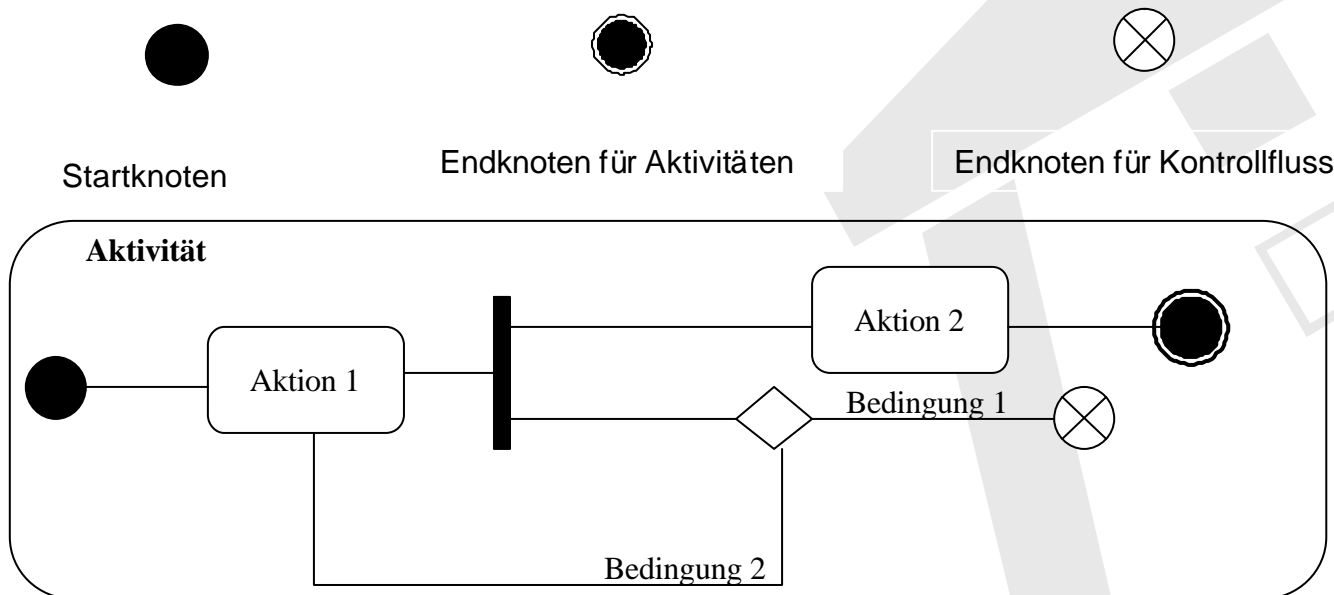


e) Kontrollelemente :

Aus einem Startknoten entsteht der Ablauf . Eine Aktivität kann beliebig viele Startknoten besitzen

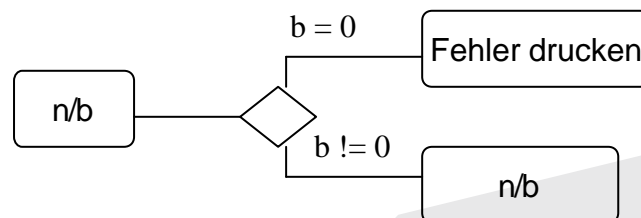
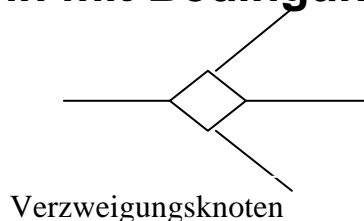
Endknoten für Aktivitäten beenden die gesamte Aktivität . Es können mehrere Endknoten für Aktivitäten existieren .

Endknoten für Kontrollflüsse markieren nur das Ende eines einzelnen Ablaufs

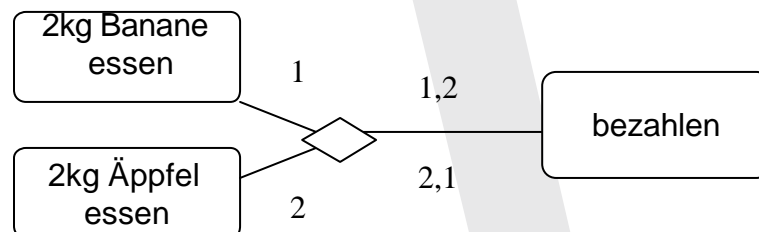
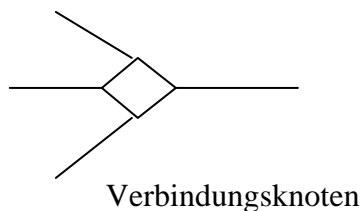


Verzweigungsknoten spaltet eine Kante in mehrere Alternativen auf .

Unter welchen Voraussetzungen welcher Zweige gewählt wird , kann mit Bedingungen festgelegt werden .



Verbindungsknoten findet keine Synchronisation an den Eingehenden Kanten statt .(keine spezifizierter Reihenfolge)



Einführung

Aktivitätsdiagramme

• Notationselemente

• Entwicklung

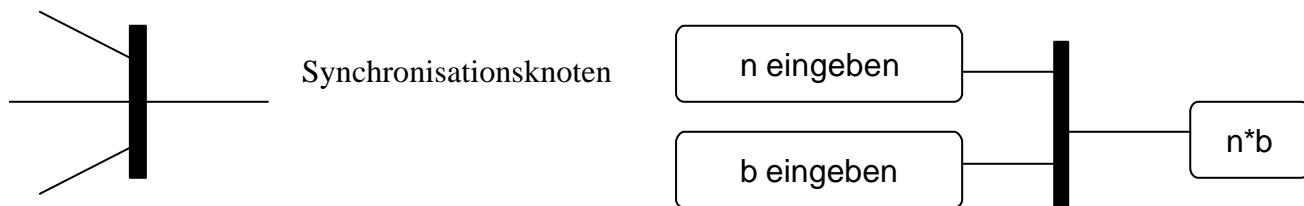
• Anwendung

Zusammenfassung

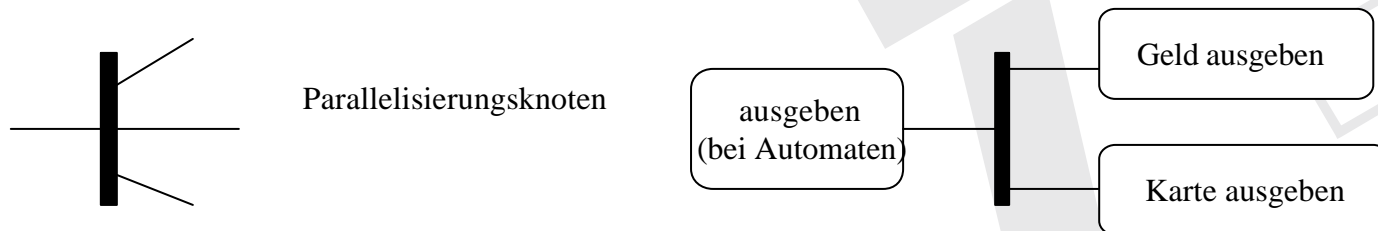
Literatur

Kontakt

Synchronisationsknoten führt gleichzeitige eingehende Abläufe zu einem gemeinsamen Ablauf zusammen .

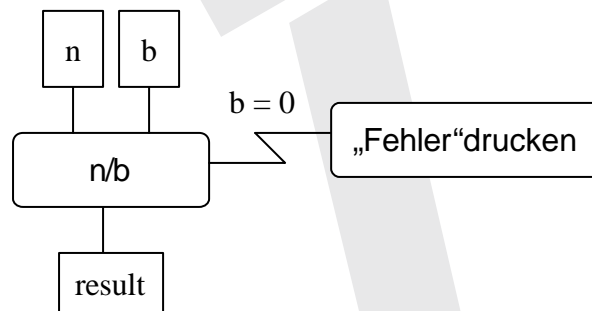
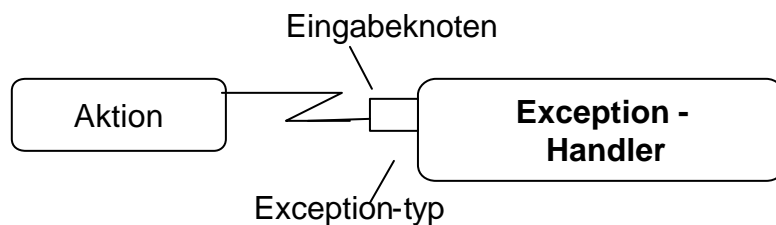


Parallelisierungsknoten teilt der eingehenden Ablauf in mehrere parallele Abläufe auf .



f) Exception-Handler :

Ein Exception - Handler bearbeitet eine vordefinierte Ausnahme , die während der Ausführung einer Aktion auftritt .



Einführung

Aktivitätsdiagramme

▸ Notationselemente

▸ Entwicklung

▸ Anwendung

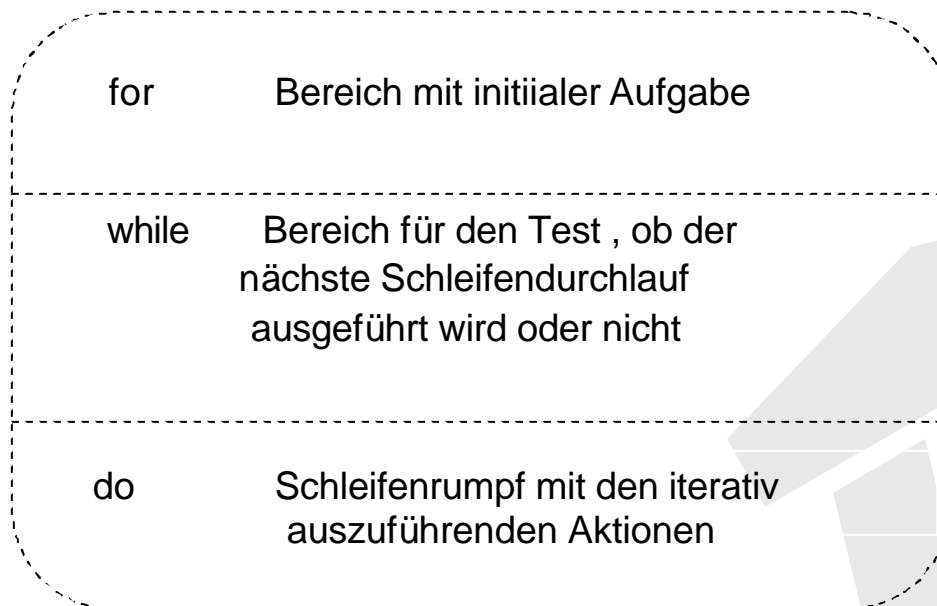
Zusammenfassung

Literatur

Kontakt

Schleifenknoten

Ein Schleifenknoten ist ein Notationselement für Schleifen in Aktivitäten . Er besteht aus einem for- , while- und do-Bereich .



Einführung

Aktivitätsdiagramme

• Notationselemente

• Entwicklung

• Anwendung

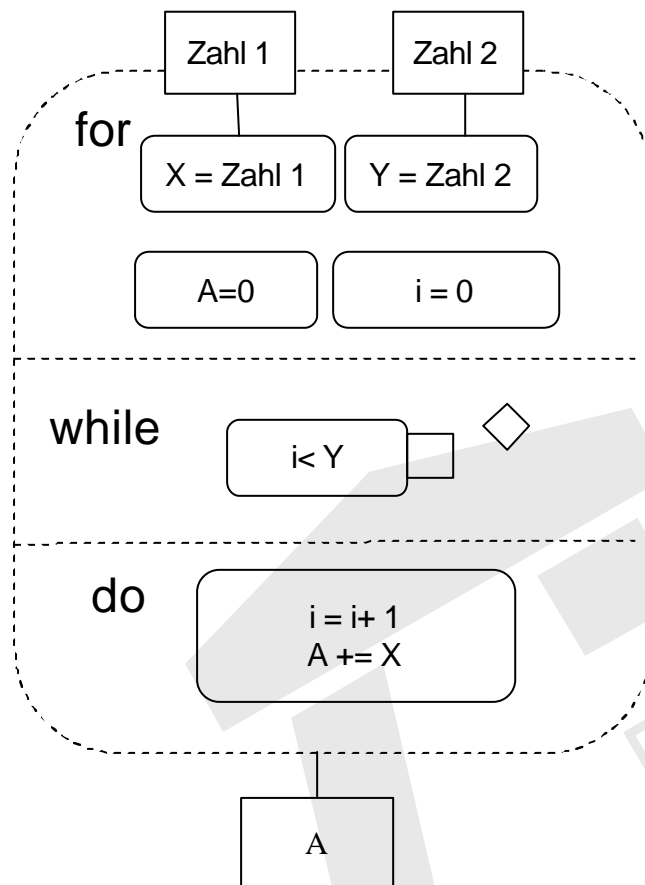
Zusammenfassung

Literatur

Kontakt

Beispiel :

```
int Mul(int X ,int Y){  
    int A = 0 ;  
    for(int i = 0; i<Y; i++){  
        A += X;  
    }  
    return A ;  
}
```

**Schleifenknoten**

Einführung

Aktivitätsdiagramme

♦ Notationselemente

♦ Entwicklung

♦ Anwendung

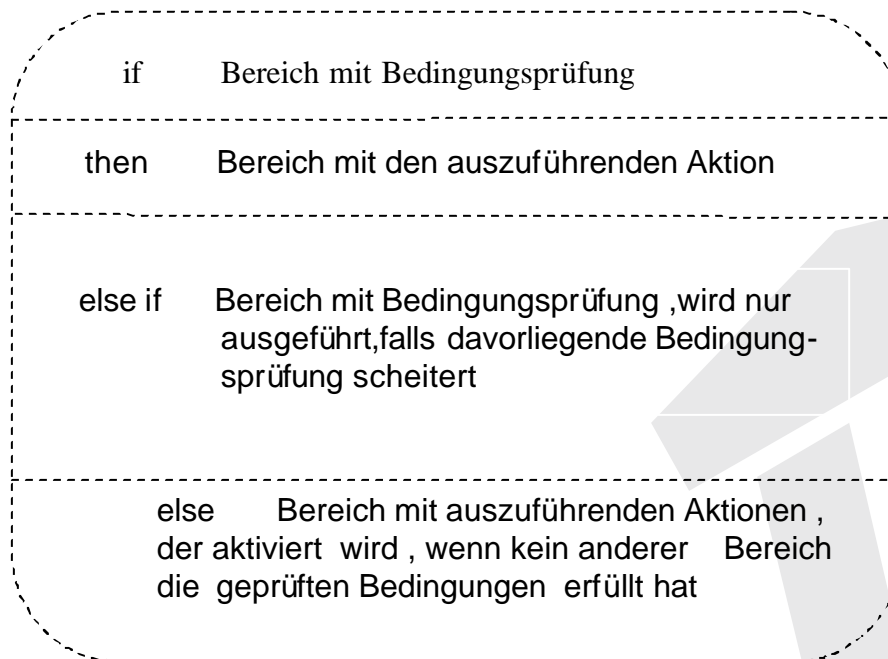
Zusammenfassung

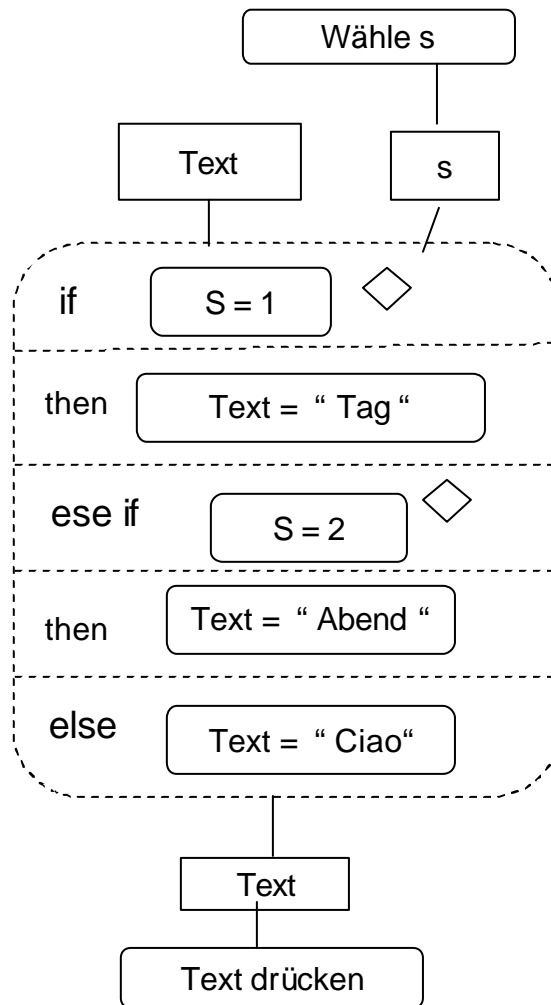
Literatur

Kontakt

Entscheidungsknoten

Ein Entscheidungsknoten modelliert, dass unter verschiedenen Bedingungen verschiedene Abläufe initiiert werden.



**Beispiel :**

```
void Ausdruck ( s ) {  
    String Text ;  
    if(S=1){  
        Text = new String ( " Tag " ) ;  
    }else if (S=2) {  
        Text = new String ( " Abend " ) ;  
    }else{  
        Text = new String ( " Ciao" ) ;  
    }  
    System.out.println(Text ) ;  
}
```

2.2 Anwendung

I Geschäftsprozessmodellierung :

Neben der Modellierung von Geschäftsprozessen stellen Aktivitätsdiagramme mächtige Notationselemente zur Verfügung. Wegen ihrer klaren grafischen Repräsentation eignen sie sich gut zur Optimierung der Prozesse und Zuordnung von Verantwortlichkeiten.

II Beschreibung von USE- Cases :

Wenn man das Verhalten von Use - Cases mit Aktivitätsdiagrammen beschreibt, ist es gut lesbar und nachvollziehbar.

III Implementierung einer Operation :

Aktivitätsdiagramme bieten die Möglichkeit (wie Struktogramme) mit Hilfe von Aktivitäten Beschreibungen ähnlich einer Programmiersprache zu modellieren. Sie stellen Code in Diagrammen klarer dar.

Einführung

Aktivitätsdiagramme

• Notationselemente

• Entwicklung

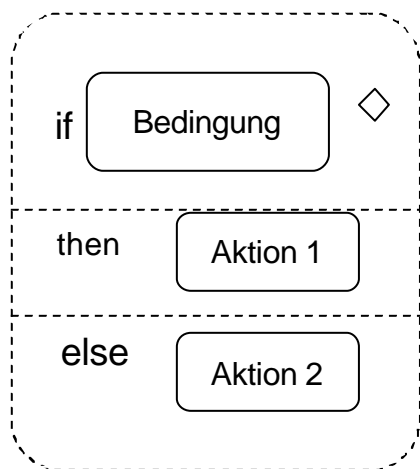
• Anwendung

Zusammenfassung

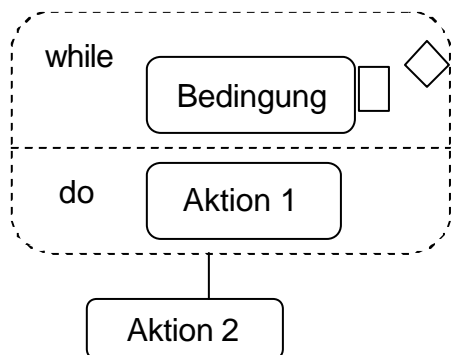
Literatur

Kontakt

Beispiel der Implementierung einer Operation : Modellierung von Struktogrammen und Java-Code

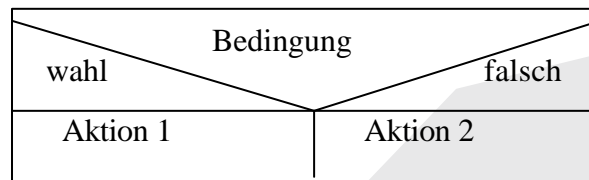


Aktivitätsdiagramm UML 2.0



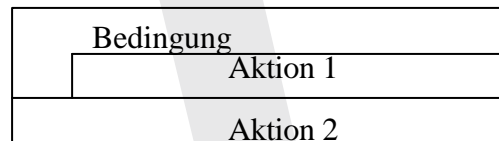
Aktivitätsdiagramm UML 2.0

```
Java Code : if ( Bedingung ) {  
    Aktion 1  
} else {  
    Aktion 2  
}
```



Struktorelementen

```
Java Code : while ( Bedingung ) {  
    Aktion 1  
}
```

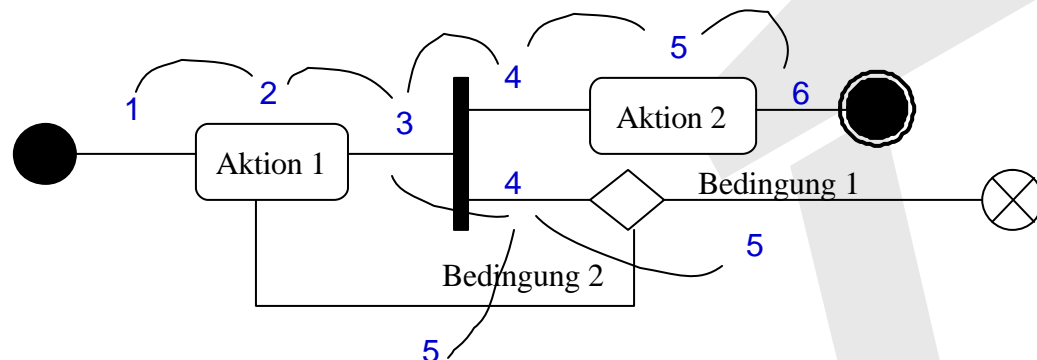


Struktorelementen

2.3 Entwicklung von UML 2

Änderung 1: Den Aktivitätsdiagrammen der UML 2 liegt eine aus den Petri-Netzen entlehnte Token - Semantik zugrunde , um die komplexen und schwer durchdringbaren Verhältnisse von nebenläufigen Abläufen zu erklären .

Token-Konzept : Mit Token werden Marken oder Staffelstäbe vorgestellt , die logisch den Punkt anzeigen , wo sich der Ablauf befindet . Jeder Punkt in einem Ablauf muss nummeriert werden .



UML 1.x : Aktivitätsdiagramme entsprechen in der Struktur den Zustandsdiagrammen .

Einführung

Aktivitätsdiagramme

♦ Notationselemente

♦ Entwicklung

♦ Anwendung

Zusammenfassung

Literatur

Kontakt

Änderung 2 : Um parallele Abläufe zu starten , sind mehrere Startknoten erlaubt . Es gibt unterschiedliche Endknoten (für Aktivitäten und für Abläufe) zur Beendigung .

UML 1.x : In einem Aktivitätsdiagramm ist nur ein Anfangszustand und ein Endzustand erlaubt .

Änderung 3 : Die UML 2 führt Objektknoten ein , um der Objektflüsse besser zu modellieren .

UML 1.x : Es gibt Objektzustände , die semantisch äußerst vage definiert sind .

Änderung 4 : In UML 2 werden neue Notationselemente ergänzt .

Beispiel : Schleifenknoten

Entscheidungsknoten

Exception-Handler u.s.w

Einführung

Aktivitätsdiagramme

• Notationselemente

• Entwicklung

• Anwendung

Zusammenfassung

Literatur

Kontakt

3 Zusammenfassung :

Wegen der Entwicklung der UML 2 entsteht ein vollständig überarbeitetes Aktivitätsdiagramm . Das Aktivitätsdiagramm ist ein mächtiges Hilfsmittel und entkoppelt Zustandsautomaten . Es besitzt mehr Eigenschaften als früher .

Token-Konzept :

Die Aktivitätsdiagrammen entlehnen eines logisches Token-Konzept , der Ablauffluss, inklusive Parallelisierung , Zusammenführung , Threading und Objektfluss geschaffen werden .

Aktivität :

In der UML 2 dürfen die Aktivitäten beliebig geschachtelt sein und durch Aktionen aufgerufen werden . Es ist möglich, gleichartiges, mehrmals beschriebenes Verhalten an genau einer Stelle zu definieren und an beliebigen anderen Stellen einzubinden. D.h. Aktivitäten können verfeinert und vergrößert werden

Einführung

Aktivitätsdiagramme

• Notationselemente

• Entwicklung

• Anwendung

Zusammenfassung

Literatur

Kontakt

Kontrolle der dynamischen Abläufe :

Die Aktivitätsdiagramme bieten spezielle strukturierte Knoten (Schleifenknoten /Entscheidungsknoten) und Exception-Handler an , um alle aus den höheren Programmiersprachen bekannten Primitiven durch eigenständige Notationselemente zu modellieren .

Beispiel : Schleifen , Selektionen (Einfach- und Mehrfachselektionen) und Ausnahmebehandlungen .

Es gibt neuen kleinere Ergänzungen, mit denen eine bessere Kontrolle und auch Darstellung der dynamischen Abläufe möglich ist.

4 Literatur

***M.Jeckle,C.Rupp,J.Hahn,B.Zengler,S.Queins:UML 2
glasklar,Hanser,2003***

