

# Zustandsdiagramme

Ein Zustandsdiagramm stellt einen endlichen Automaten in einer UML Sonderform grafisch dar.

# Was bildet es ab?

Ein Zustandsdiagramm beschreibt die möglichen Folgen von Zuständen die ein Modell-Element (Objekt einer bestimmten Klasse) während seiner Lebensdauer oder während es eine bestimmte Operation ausführt durchlaufen kann.

Des weiteren wird beschrieben, aufgrund welcher Ereignisse ein Zustandsübergang stattfindet.

# Wofür sind Zustandsdiagramme geeignet?

- Um das Verhalten von einzelnen Komponenten oder eines Systems darzustellen.
- Um zulässige Nutzung der Schnittstelle eines Systems zu spezifizieren.

(Welche Ereignisse verursachen Zustandsübergang)

# Die Elemente

- Zustände
- Entscheidungsknoten
- Transitionen
- Aktivitäten



# Zustand

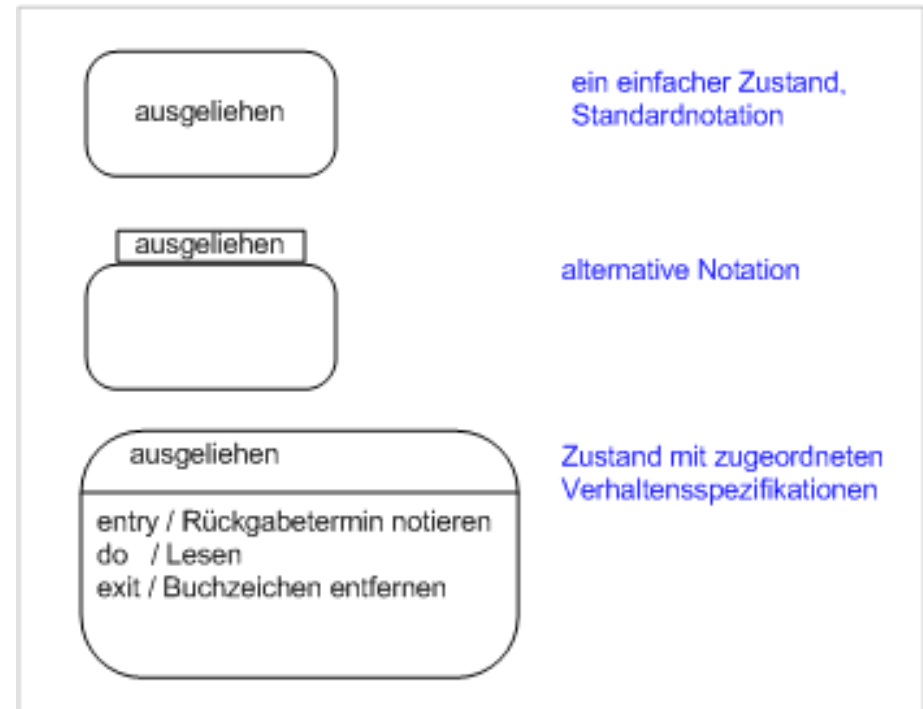
Zustände = Knoten

Entweder einfacher Zustand oder Pseudo-Zustand wie der Startzustand.

Ein Pseudo-Zustand kann nicht dauerhaft in diesem Zustand bleiben und dort auch keine Wertbelegung möglich

Startzustand (beginn des Zustandsdiagramms, keine eingehenden Transitionen nur eine ausgehende)

Endzustand (keine ausgehenden Transitionen)

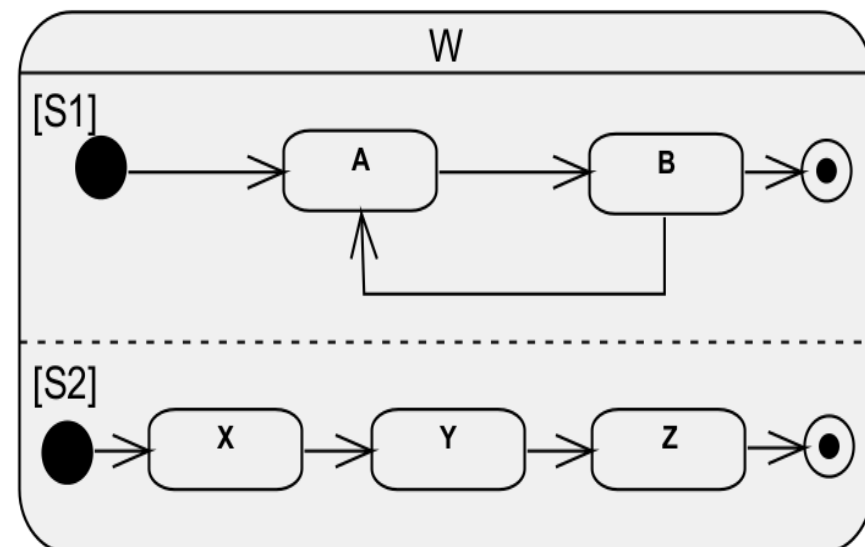
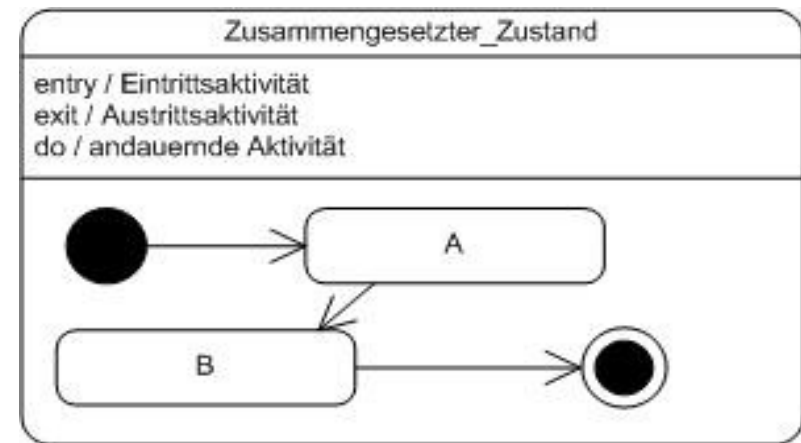


# Komplexe Zustände

komplexe Zustände  
= geschachteltes  
Zustandsdiagramm

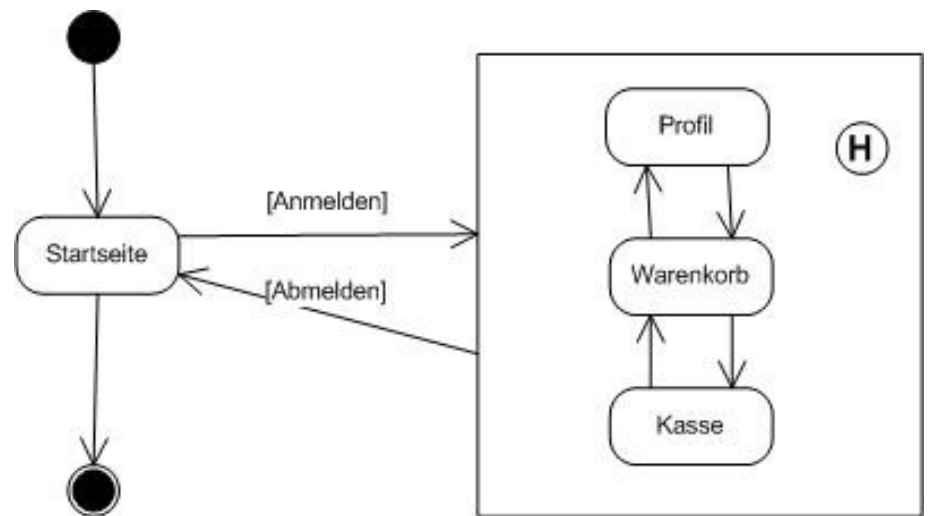
Trotzdem nur einer der  
Subzustände aktiv(A oder B)

Aber Teilung in mehrere  
Regionen (mit mehreren  
Start- und Endzuständen)  
erlaubt Parallel aktive  
Zustände.



# Historie Zustände

- Merken sich internen Zustand in einem komplexen Zustand von dem die letzte Transition ausgegangen ist
- Zu einem späteren Zeitpunkt kann zu diesem Zustand über Transitionen aus übergeordneten Zuständen zurückgekehrt werden
- Flacher History-Zustand  $H$  merkt sich eine Ebene
- Tiefer History-Zustand  $H^*$  sichert Zustände über die gesamte Schachtelungstiefe hinweg
- Im Beispiel befindet sich System auf der Seite von wo es sich bei letzten Abmelden befand.



# Entscheidungsknoten

## Entscheidungsknoten

if Abfrage mit alternativen  
Transitionen

## Parallelisierungsknoten

Spaltet Kontrollfluss in mehrere  
parallele Zustände auf

## Synchronisierungsknoten

Führt Kontrollfluss von mehreren  
parallelen Zuständen  
zusammen.



Startzustand



Endzustand



Synchronisation  
oder Gabelung



Kreuzung oder  
Entscheidung



Eintrittspunkt



Austrittspunkt



flache Historie

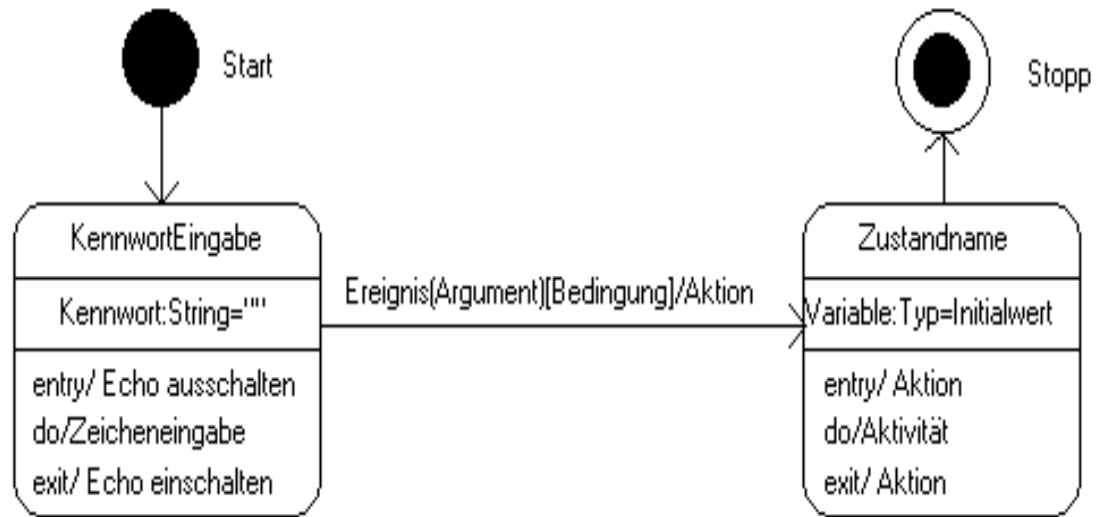


tiefe Historie



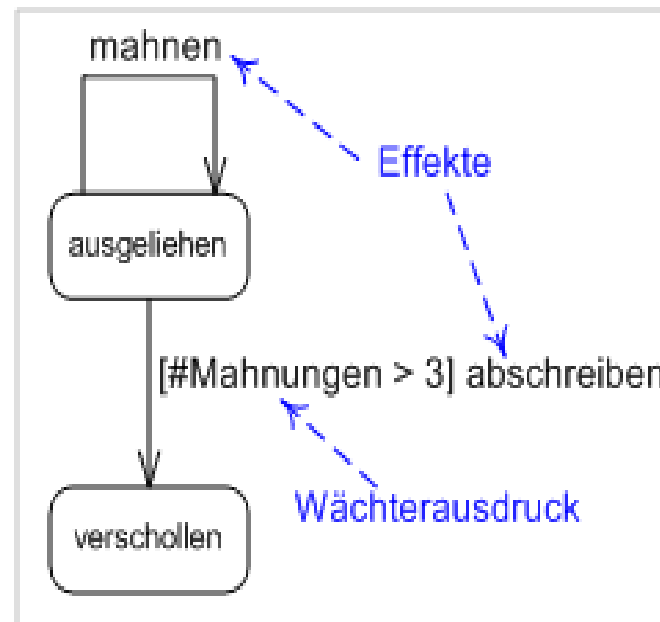
# Ereignisse

- entry
- do
- exit
- Oder selbst festgelegte



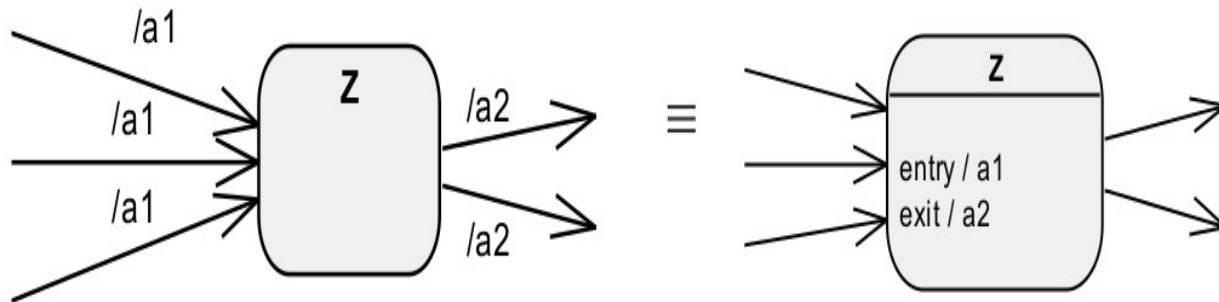
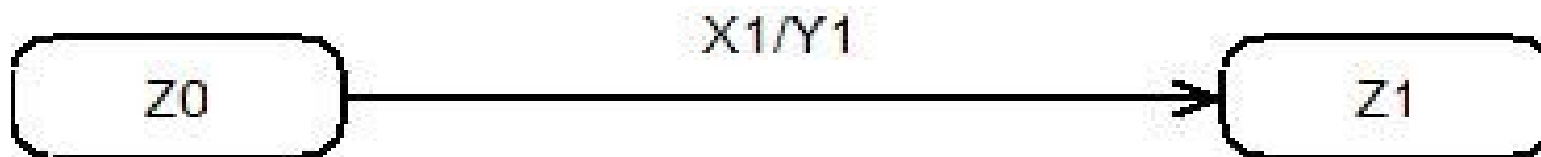
# Transition = Zustandsübergang

- Ausgelöst durch Ereignis und verbindet einen Quell- und einen Zielknoten.
- Innere Transition falls Zustand nicht verändert wird und keine entry, exit Aktivitäten ausgeführt werden.
- Selbsttransition falls Zustand nicht verändert wird und entry und exit Aktivitäten ausgeführt werden.
- Bedingungen für Zustandsübergang angeben möglich (Wächterausdruck)



# Schreibweise / Notation

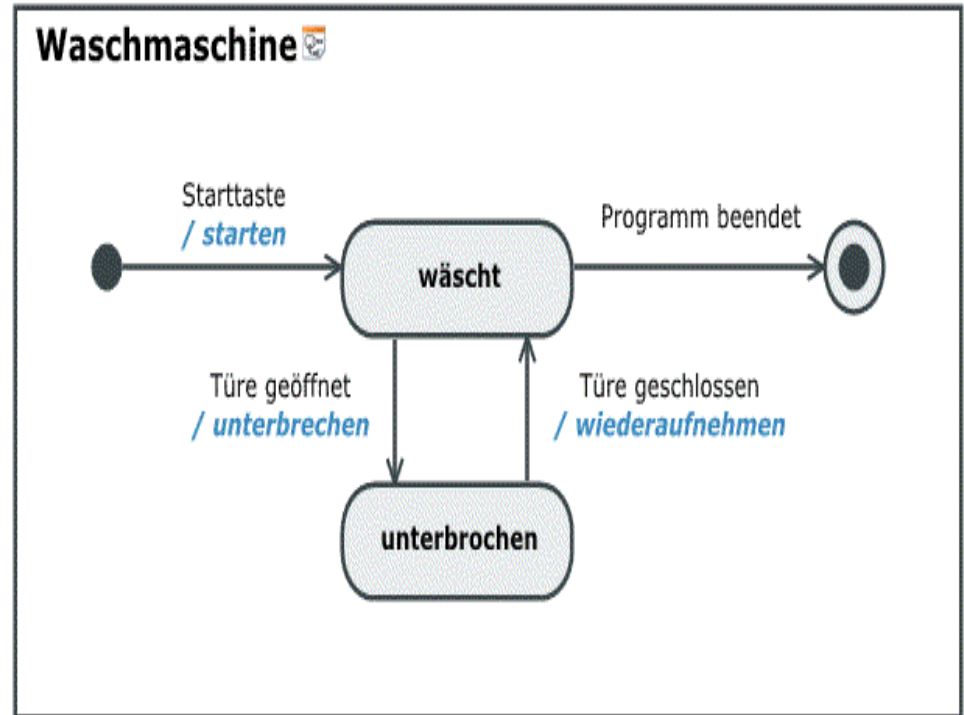
Ereignis(Argumente) [Bedingung] / Aktivität(en)



# Aktivitäten

Werden in Zuständen ausgeführt oder bei einer Transitionen wenn das jeweilige Ereignis eintritt.

Aktivität zum Beispiel Variablenwert inkrementieren.



# Geeignete Metriken

## Größen Metriken:

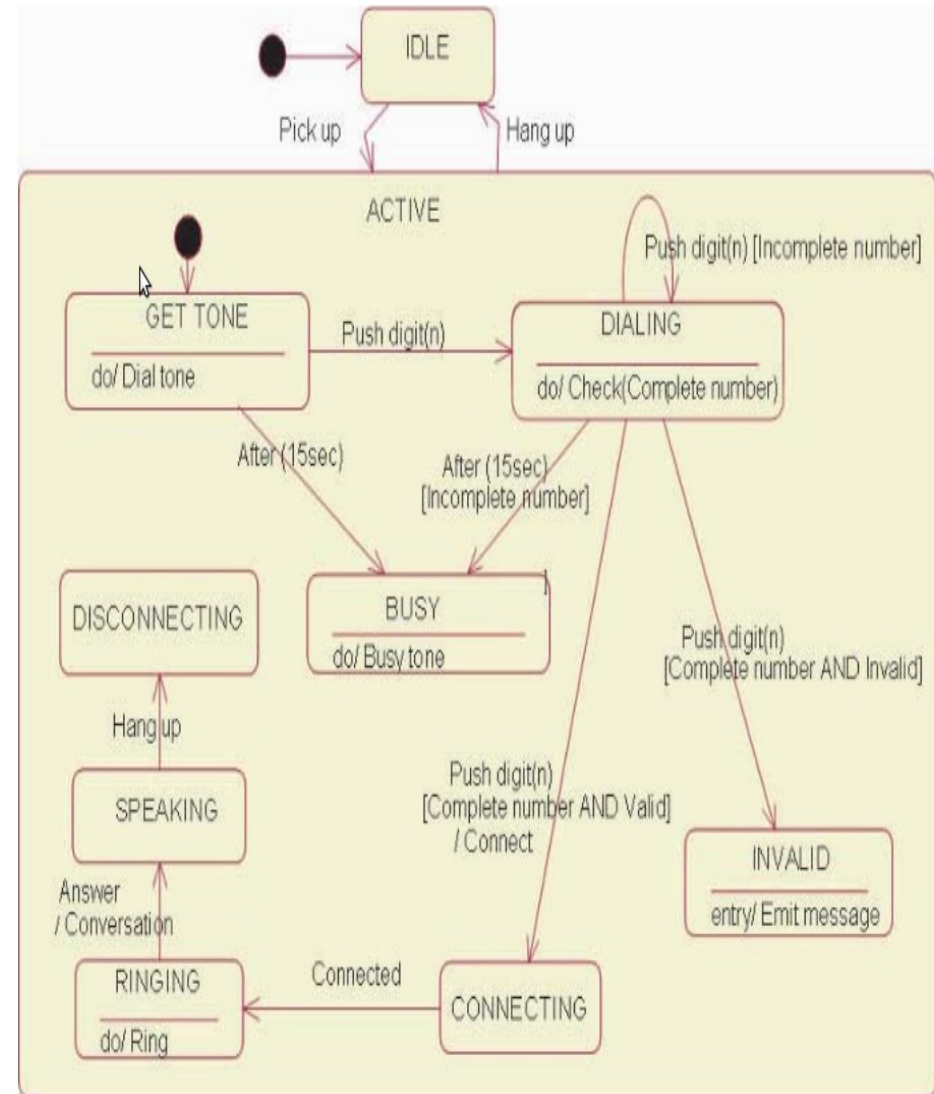
- Gesamtanzahl der Zustände einer Klasse
- Gesamtanzahl der Aktivitäten einer Klasse
- Gesamtanzahl der entry Aktivitäten
- Gesamtanzahl der exit Aktivitäten
- Gesamtanzahl der do Aktivitäten
- Gesamtanzahl der einfachen (inklusive einfachen Zustände in zusammengesetzten Zuständen) Zustände
- Gesamtanzahl der zusammengesetzten Zustände
- Gesamtanzahl der Ereignisse
- Gesamtanzahl der Wächterausdrücke

## Strukturelle Komplexität Metriken:

- Gesamtanzahl der Transitionen (Äußere Transitionen, innere Transitionen, Selbsttransitionen, Start- und Endtransitionen)
- Cyclomatic Number of McCabe:  
$$|\text{einfache Zustände}| - |\text{Gesamtanzahl Transitionen}| + 2$$

# Beispiele

- Gesamtanzahl der Transitionen: 13
- Gesamtanzahl der Ereignisse: 11
- Gesamtanzahl Zusammengesetzter Zustände: 1
- Gesamtanzahl Einfacher Zustände: 9
- Gesamtanzahl Wächterausdrücke: 4
- Gesamtanzahl entry Aktivitäten: 1
- Gesamtanzahl exit Aktivitäten: 0
- Gesamtanzahl der do Aktivitäten: 4
- McGabe Number:  $|9 - 13 + 2| = 2$



# Geeignete Smells

- No Incoming = keine eingehende Transition
- Unnamed State = Zustand unbenannt

# Geeignete Refactorings

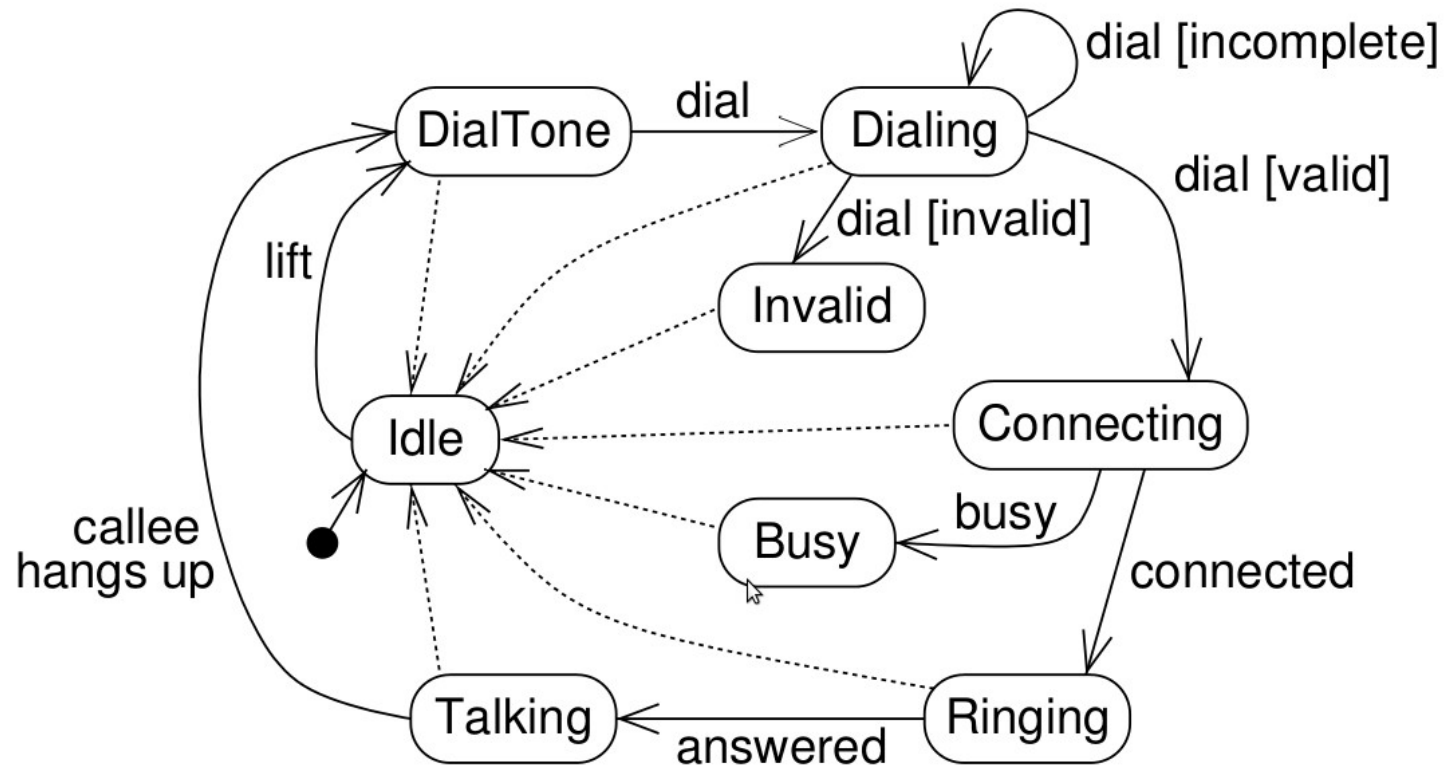
- Fold Incoming/Outgoing Actions (gleiche Aktionen in Zustand durch entry/exit hineinziehen)
- Unfold Entry/Exit Action (exit/entry Aktionen aus Zustand herausholen)
- Fold Outgoing Transitions (gleiche Transitionen aus Zuständen in zusammengesetzten Zustand durch eine Transition aus zusammengesetzten Zustand ersetzen)
- Unfold Outgoing Transitions (Transition aus zusammengesetzten Zustand ersetzen durch Transitionen aus allen inneren Zuständen)
- Zustand in zusammengesetzten Zustand einfügen
- Zustand aus zusammengesetzten Zustand herausholen
- Zustand umbenennen
- Isolierten Zustand entfernen
- Innerer Transitionen hineinziehen falls keine exit oder entry Aktivitäten vorhanden sind
- Zustände vereinigen
- Zusammengesetzten Zustand bilden (gruppieren)



# Refactoring Constraints

- Refactorings können nur durchgeführt werden wenn bestimmte constraints erfüllt sind.
- Constraints stellen sicher das sich Verhalten durch Refactoring nicht verändert.

# Beispiel



# Beispiel

1. Erstelle zusammengesetzten Zustand „Active“ der ganzes Diagramm umgibt.
2. Der Startzustand und der „Idle“ Zustand werden aus dem zusammengesetzten Zustand herausgeholt.
3. Die verschiedenen „hangup“ Transitionen (gestrichelte Linien) zusammenfassen und aus „Active“ herausführen.
4. Die „Lift“ Transition von „Idle“ nur noch bis zur Grenze von Zustand „Active“ führen und im Zustand „Active“ einen neuen Startzustand erstellen der direkt in Zustand „Dial Tone“ übergeht.

Begründungen:

1. Zusammengesetzter Zustand verändert Verhalten nicht.
2. Da Zustand „Active“ keine entry/exit Aktionen hat kann „Idle“ herausgeholt werde ohne das sich Reihenfolge in der Aktionen ausgeführt werden ändert.
3. Äquivalente ausgehende Transitionen „hangup“ (gleiche Zielknoten) können zusammengefasst werden.
4. Ersetzen der „lift“ Transition ist erlaubt, da nur von „Idle“ Transition in Zustand „Active“ führt.

