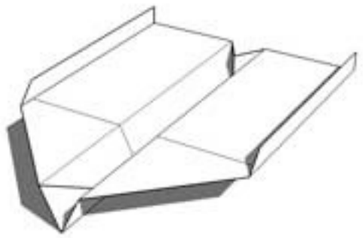
EDV & Multimedia
Interaktionsdesign

12 – Modellierung

Prof. Dr. Jochen Koubek

23. Januar 2012



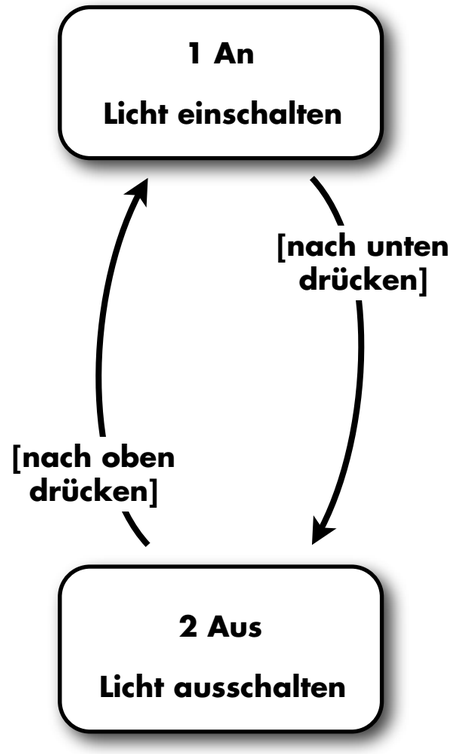
Modell

Ein Modell ist die Abbildung eines Weltausschnittes in einem anderen Medium (z.B. Papier, Begriffe, Mathematik, Datenstrukturen und Algorithmen)

Ein Modell konzentriert sich auf bestimmte Merkmale der Vorlage (z.B. Aussehen, Funktion, Verhalten) und lässt andere weg.

Jedes Modell wird zu einem bestimmten Zeitraum (wann?) von jemandem (von wem?) für jemanden (für wen?) aus einem Grund (warum?) für Anwendungsfälle (wozu?) eines bestimmten Zeitraum (für wann?) konstruiert.

Endliche Automaten



Lichtschalter

Ein endlicher Automat ist ein Modell für das Verhalten eines interaktiven Systems.

Zu jedem Zeitpunkt befindet er sich in genau einem **Zustand** von einer begrenzten (endlichen) Menge an möglichen Zuständen.

Je nach **Ereignis** (z.B. Eingabe, Messwert, Zeit) wechselt er in einen anderen Zustand (**Zustandsübergang)**.

In jedem Zustand werden verschiedene **Aktionen** ausgeführt, die das **Verhalten** des Systems bestimmen.

Zustandsdiagramm (UML-Notation)

http://de.wikipedia.org/wiki/Zustandsdiagramm_(UML)

http://en.wikipedia.org/wiki/UML_state_machine

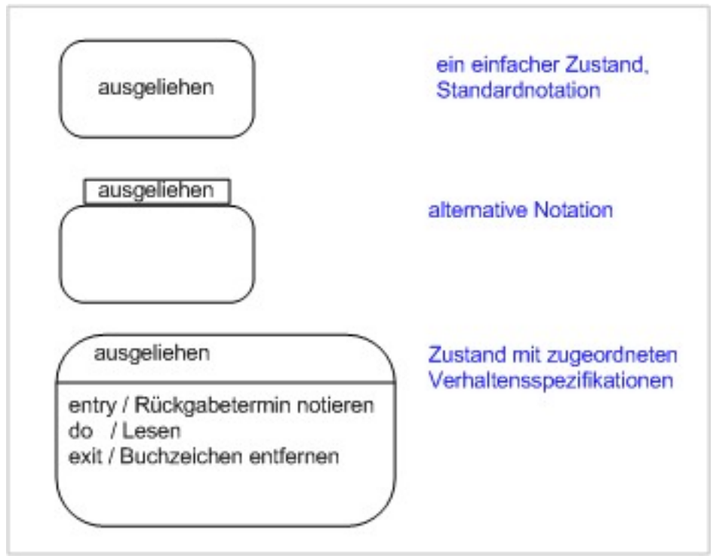
Startzustand

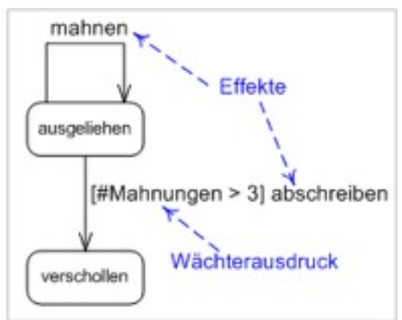
Endzustand (optional)

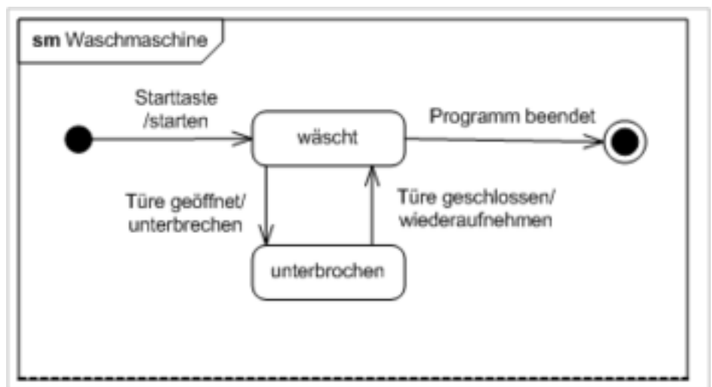
Zustand

Verhalten
entry | do | exit

Übergang (Transition)
[*Wächterausdruck*] Effekt





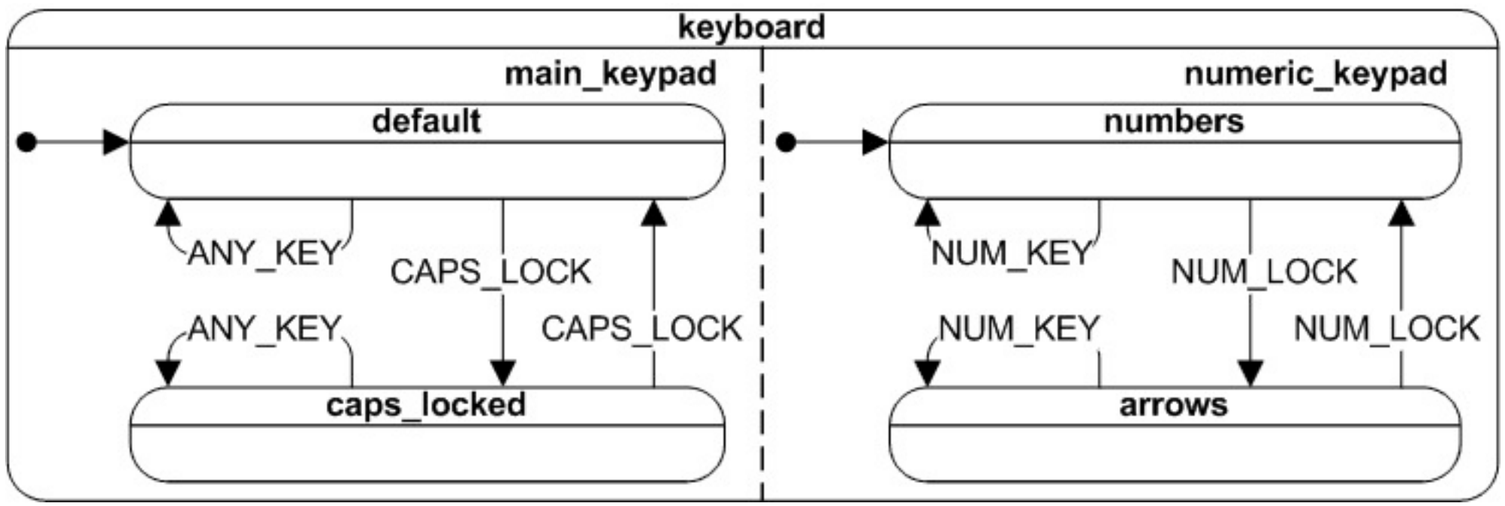


Übung

Modellieren Sie diesen Auto-Lichtschalter mit einem Zustandsdiagramm

Wie ist es mit diesem?

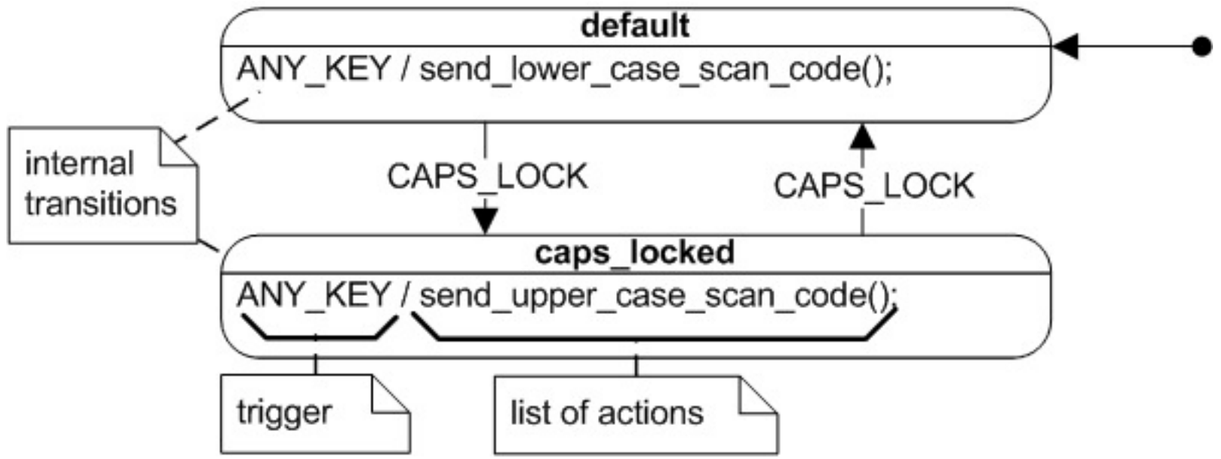
Unabhängige Teilsysteme (Orthogonale Regionen)



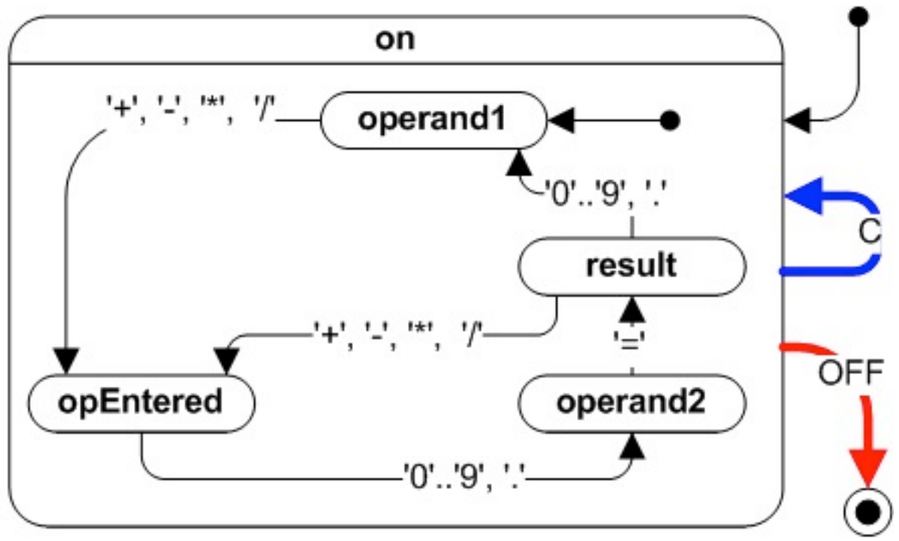
http://en.wikipedia.org/wiki/UML_state_machine

Interne Übergänge

Was ist mit der Helligkeitsregelung?

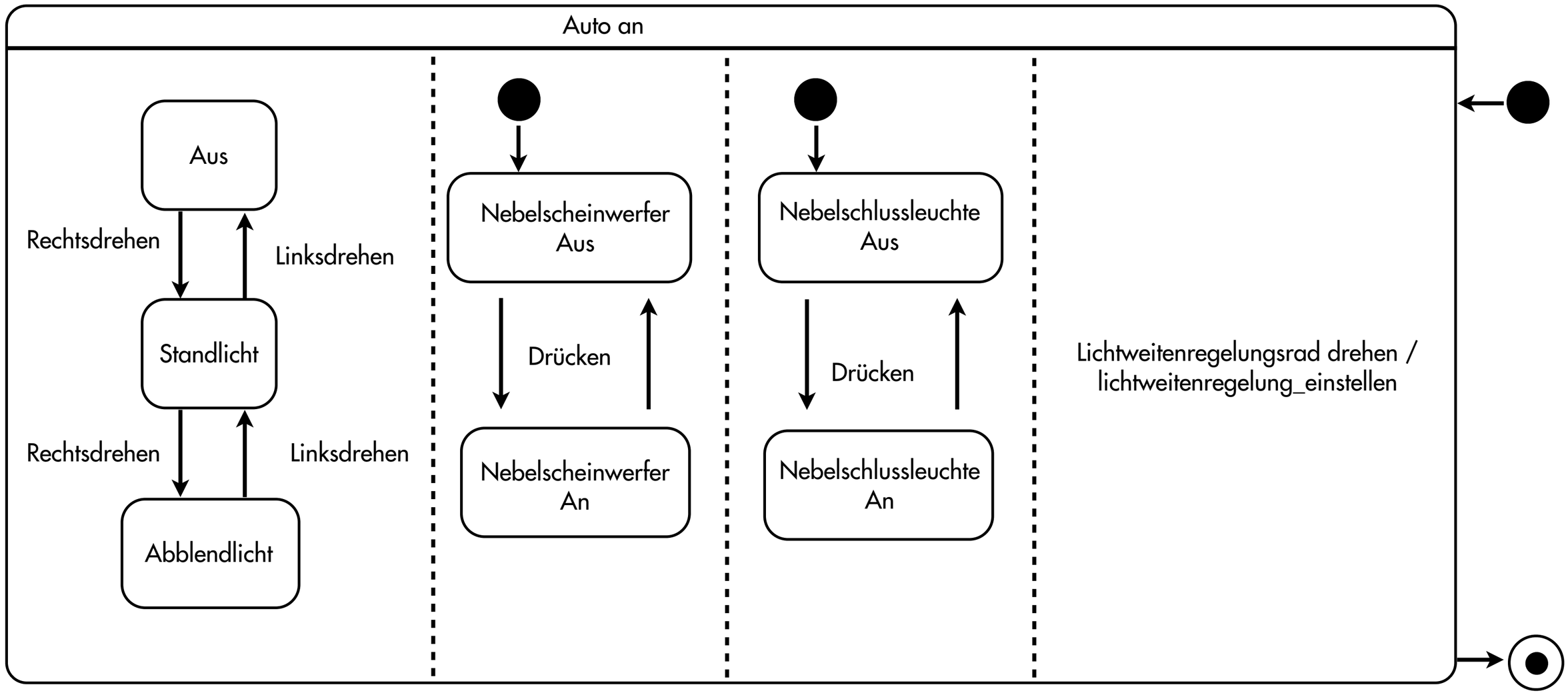


Zustandshierarchien



Ein **Zusammengesetzter Zustand** bestehet aus einem Oberzustand und mehreren Unterzuständen. Die Rechentasten des Taschenrechners funktionieren nur im eingeschalteten Zustand.

Auto-Lichtschalter



Implementierung von Endlichen Automaten

Im Rahmen unserer Prototypen arbeiten wir mit

```
if (zustand == 1) {… } else if (zustand == n)
```

bzw.

```
switch (zustand) {case 1: … case n: … }
```

Wobei der aktuelle Zustand z.B. in einer Integer-Variable gespeichert wird.

Obere Zustandshierarchien

Processing bietet mit den Funktionen `setup()`, `draw()` und `stop()` Rahmen für entry-, do- und exit-Verhalten des Hauptprogramms.

Auto-Lichtschalter

```
int zustand;

void setup() {
  zustand = 1;
  println("System an");
  background(0);
}

void draw() {

/****************************************
Interne Übergänge verarbeiten ohne Zustände
****************************************/

//...

/*******************************
Teilsystem 1: Hauptlicht
*******************************/

  if (zustand == 1) {
    background(0);
    println("Licht aus");
    if(keyPressed){
      if (key=='r'){
        zustand = 2;
        println("Zustand wechseln: 1 -> 2");
       }
    }
  }

  if (zustand == 2) {
    background(128);
    println("Standlicht an");
    if(keyPressed){
      if (key=='R'){
        zustand = 3;
        println("Zustand wechseln 2 -> 3");
       }
      if (key=='l'){
        zustand = 1;
        println("Zustand wechseln 2 -> 1");
       }
    }
  }

  if (zustand == 3) {
    background(255);
    println("Abblendlicht an");
    if(keyPressed){
      if (key=='L'){
        zustand =2;
        println("Zustand wechseln 3 -> 2");
       }
    }
  }

/*******************************
Teilsystem 2: Nebelscheinwerfer
*******************************/
//...

}

void stop() {
  zustand = 1;
  println("System aus");
}
```

Übung

Modellieren Sie den Mensa-Kartenautomaten