

# Regelbasierte Systeme DAN+WI

02.12.2010

Thomas Galliker

## Begriffe

- ⇒ Business Rules = Geschäftsregeln
- ⇒ BRMS = GR-Managementsystem

## Vorteile

- ⇒ Code getrennt von Geschäftsregeln.
- ⇒ Regeln können vom Business undefiniert werden.
- ⇒ System erlaubt keine Interpretationsspielräume und ist somit fairer als ein Mensch.
- ⇒ Regeln können jederzeit neuen Gegebenheiten angepasst werden.

## Nachteile

- ⇒ Nicht alle Fälle können automatisch mit Regeln beantwortet werden.

## Aktionsregel

- ⇒ Aktion wird ausgeführt, u.U. Wissen verändert: WENN X, DANN führe P3 aus
- ⇒ Reihenfolge spielt eine Rolle im Gegensatz zu Folgeungsregel

## Folgerungsregel

- ⇒ Vorwärts WENN X eintritt, DANN... Säugtier
- ⇒ Rückwärts WENN Säugtier, DANN hat Haare

## Beispiel Folie 11

Wir wissen nur, dass das Tier ein Gepard ist.

⇒ TG ⇒ Gepard, deshalb Fleischfresser + gebraun + Flecken

⇒ weitere Regeln prüfen

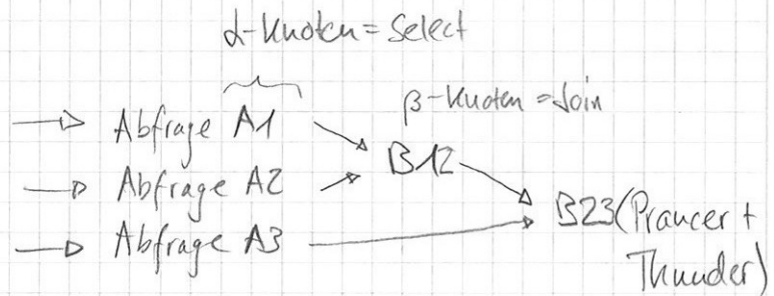
## Beispiel: Tüte packen

- Warenkorb:
  - Kartoffelchips
  - 7Fl. Cola
  - 1x Pampe

- Nächster Schritt "P2"  $\Rightarrow$  1 Schritt weiter
- P3: Neue Tüte  $\Rightarrow$  Artikel packen

## Beispiel: Rennpferde

Wenn X ein Pferd  
X ist Elternteil von Y  
Y ist schnell  
dann ist X wertvoll



- Mit Assert werden neue Datensätze eingefügt

## BRMS Eigenschaften

- Rule Engines
- Regeleditoren
- Versionsverwaltung
- Test Support

## HELLO WORLD BEISPIEL

Rule Base

- Regelsätze

Working Memory

- Enthält Objekte
- Manipulierbar

Agenda

- Buchführung der Regeln

} W.M. und A. sind Teil einer Session

- 1) Message Objekt erstellen:
- 2) Message/Status setzen
- 3) Mit insert(Message) in Working Memory einfügen
- 4) Regeln ausführen mit FireAllRules();
- 5) Regel 1 feuert  $\Rightarrow$  DANN-Teil wird in Agenda eingefügt

## Komponenten eines regelbasierten Systems

- Wissensbasis: Fachwissen (langfristig), Fallwissen (pro Fall), Zwischenergebnisse
- Problemlösungskomponente: Rule Engine,
- Erklärungskomponente: Zusatzinfo, Begründung der Fallentscheidung
- Wissenserwerbskomponente: Editoren, welche vom Business zum Erstellen von Regeln genutzt wird.

## Definition von Regeln

```
rule "Rule Name"
```

```
  when
```

```
    t: Tier ( hatHaare == ThreeValBool.TRUE,  
              istSäugetier := ThreeValBool.TRUE )
```

```
  then
```

```
    t.setIstSäugetier (ThreeValBool.TRUE);  
    update (t); // Regel neu auswerten
```

```
end
```

# Woher kommen die Regeln

18.12.2020

- ⇒ Analyse von Geschäftsprozessen
- ⇒ Expertenwissen

## Kostenmatrix

	Predicted Class	
	+	-
Actual Class	+	-1
	-	1

Jede Entscheidung liefert...  
▷ Gewinn (=neg. Kosten)  
▷ Verlust

Model 1	Predicted Class	
	+	-
Act. Class	+	150
	-	60

Model 2	Predicted Class	
	+	-
Actual Class	+	250
	-	5

Accuracy =  $(150+250) / (150+250+60+5) = 0,8$   
Cost =  $-1 \cdot 150 + 40 \cdot 100 + 1 \cdot 60 + 0 \cdot 250 = 3910$