

Seminar 18.410: Programmiersprachenkonzepte • SS2004

# Agent-Oriented Programming (AOP)

Daniel Doberenz

Universität Hamburg • Fachbereich Informatik

# Inhalt

# Inhalt

## 1. Agententechnologie

# Inhalt

1. Agententechnologie
2. Einsatzgebiete

# Inhalt

1. Agententechnologie
2. Einsatzgebiete
3. Unterschiede zu OOP

# Inhalt

1. Agententechnologie
2. Einsatzgebiete
3. Unterschiede zu OOP
4. AOP Framework

# Inhalt

1. Agententechnologie
2. Einsatzgebiete
3. Unterschiede zu OOP
4. AOP Framework
5. mental state

# Inhalt

1. Agententechnologie
2. Einsatzgebiete
3. Unterschiede zu OOP
4. AOP Framework
5. mental state
6. AGENT0



# Inhalt

1. Agententechnologie
2. Einsatzgebiete
3. Unterschiede zu OOP
4. AOP Framework
5. mental state
6. AGENT0
7. Interpreter

# Inhalt

1. Agententechnologie
2. Einsatzgebiete
3. Unterschiede zu OOP
4. AOP Framework
5. mental state
6. AGENT0
7. Interpreter

# 1. Agententechnologie

# 1. Agententechnologie

Definition Agent aus [SH090]:

# 1. Agententechnologie

Definition Agent aus [SH090]:

- Agenten können Dienste anbieten

# 1. Agententechnologie

Definition Agent aus [SH090]:

- Agenten können Dienste anbieten
- Umgebung kann unbekannt sein

# 1. Agententechnologie

Definition Agent aus [SH090]:

- Agenten können Dienste anbieten
- Umgebung kann unbekannt sein
- formale Agenten als  
„artificial agents“

# 1. artificial agents



# 1. artificial agents

Bestandteile:

# 1. artificial agents

Bestandteile:

- knowledge = Wissen

# 1. artificial agents

Bestandteile:

- knowledge = Wissen
- beliefs = Vorstellungen

# 1. artificial agents

Bestandteile:

- knowledge = Wissen
- beliefs = Vorstellungen
- capabilities = Befähigungen

# 1. artificial agents

Bestandteile:

- knowledge = Wissen
- beliefs = Vorstellungen
- capabilities = Befähigungen
- choices = Auswahl

# 1. artificial agents

Bestandteile:

- knowledge = Wissen
- beliefs = Vorstellungen
- capabilities = Befähigungen
- choices = Auswahl
- commitments = Verpflichtungen

# 1. artificial agents

Bestandteile:

- knowledge = Wissen
- beliefs = Vorstellungen
- capabilities = Befähigungen
- choices = Auswahl
- commitments = Verpflichtungen
- und wenn nötig auch andere

# 1. Möglichkeiten



# 1. Möglichkeiten

- fast alles beschreibbar

# 1. Möglichkeiten

- fast alles beschreibbar
- aber nicht immer sinnvoll

# 1. Möglichkeiten

- fast alles beschreibbar
- aber nicht immer sinnvoll
- Beispiel: Lichtschalter als Agent

# 1. Möglichkeiten

- fast alles beschreibbar
- aber nicht immer sinnvoll
  
- Beispiel: Lichtschalter als Agent
  - Lichtschalter gibt Strom weiter

# 1. Möglichkeiten

- fast alles beschreibbar
- aber nicht immer sinnvoll
  
- Beispiel: Lichtschalter als Agent
  - Lichtschalter gibt Strom weiter
  - auf unseren Wunsch

# 1. Möglichkeiten

- fast alles beschreibbar
- aber nicht immer sinnvoll
- Beispiel: Lichtschalter als Agent
  - Lichtschalter gibt Strom weiter
  - auf unseren Wunsch
  - Kommunikation über Schalterstellung

# 1. Möglichkeiten

- fast alles beschreibbar
- aber nicht immer sinnvoll
- Beispiel: Lichtschalter als Agent
  - Lichtschalter gibt Strom weiter
  - auf unseren Wunsch
  - Kommunikation über Schalterstellung
  - elektronische Beschreibung sinnvoller

# Inhalt

1. Agententechnologie
2. Einsatzgebiete
3. Unterschiede zu OOP
4. AOP Framework
5. mental state
6. AGENT0
7. Interpreter



# 2. Einsatzgebiete (1)

# 2. Einsatzgebiete (1)

- Beispiel 1: Lieferroboter

## 2. Einsatzgebiete (1)

- Beispiel 1: Lieferroboter
- neues Informatikum 2010

## 2. Einsatzgebiete (1)

- Beispiel 1: Lieferroboter
- neues Informatikum 2010
- Lieferroboter: Gofer

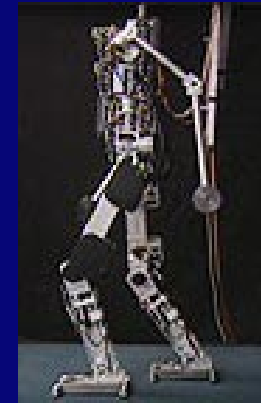
## 2. Einsatzgebiete (1)

- Beispiel 1: Lieferroboter
- neues Informatikum 2010
- Lieferroboter: Gofer
- Treffen von Gofer1 und Gofer2

## 2. Einsatzgebiete (2)



Gofer1



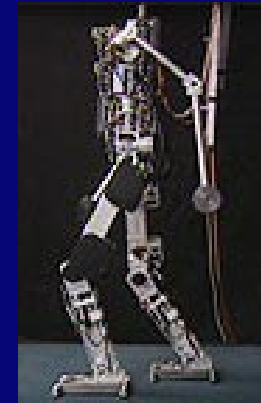
Gofer2

## 2. Einsatzgebiete (2)

Gofer1: Ich muss zu C220



Gofer1



Gofer2

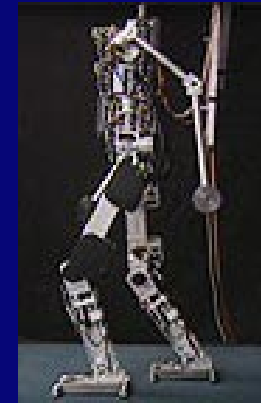
## 2. Einsatzgebiete (2)

Gofer1: Ich muss zu C220

Gofer2: Ich zu C202, was hast du vor?



Gofer1



Gofer2



## 2. Einsatzgebiete (2)

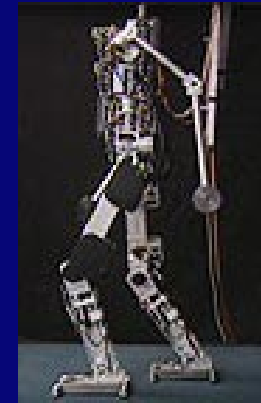
Gofer1: Ich muss zu C220

Gofer2: Ich zu C202, was hast du vor?

Gofer2: Kopieren.



Gofer1



Gofer2

## 2. Einsatzgebiete (2)

Gofer1: Ich muss zu C220

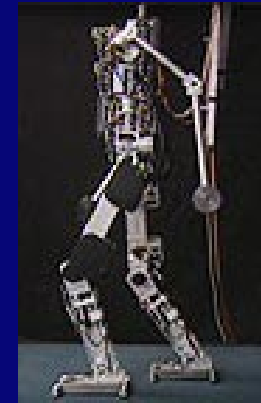
Gofer2: Ich zu C202, was hast du vor?

Gofer2: Kopieren.

Gofer1: Kannst du für mich kopieren?



Gofer1



Gofer2

## 2. Einsatzgebiete (2)

Gofer1: Ich muss zu C220

Gofer2: Ich zu C202, was hast du vor?

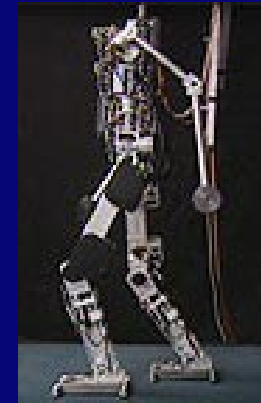
Gofer2: Kopieren.

Gofer1: Kannst du für mich kopieren?

Gofer2: Okay.



Gofer1



Gofer2

## 2. Einsatzgebiete (3)

## 2. Einsatzgebiete (3)

- Beispiel 2: Flug buchen

## 2. Einsatzgebiete (3)

- Beispiel 2: Flug buchen
- Passagier P und Angestellter C

## 2. Einsatzgebiete (3)

- Beispiel 2: Flug buchen
- Passagier P und Angestellter C
  - P: Welche Flüge Hamburg – Frankfurt?

## 2. Einsatzgebiete (3)

- Beispiel 2: Flug buchen
- Passagier P und Angestellter C
  - P: Welche Flüge Hamburg – Frankfurt?
  - C: Flug LH123 um 8:30, Flug LH124 um 10:00, Flug LH125 um ...



## 2. Einsatzgebiete (3)

- Beispiel 2: Flug buchen
- Passagier P und Angestellter C
  - P: Welche Flüge Hamburg – Frankfurt?
  - C: Flug LH123 um 8:30, Flug LH124 um 10:00, Flug LH125 um ...
  - P: Ich nehme LH123.

## 2. Einsatzgebiete (3)

- Beispiel 2: Flug buchen
- Passagier P und Angestellter C
  - P: Welche Flüge Hamburg – Frankfurt?
  - C: Flug LH123 um 8:30, Flug LH124 um 10:00, Flug LH125 um ...
  - P: Ich nehme LH123.
  - C: Der ist ausgebucht.

## 2. Einsatzgebiete (3)

- Beispiel 2: Flug buchen
- Passagier P und Angestellter C
  - P: Welche Flüge Hamburg – Frankfurt?
  - C: Flug LH123 um 8:30, Flug LH124 um 10:00, Flug LH125 um ...
  - P: Ich nehme LH123.
  - C: Der ist ausgebucht.
  - P: Ich nehme LH124.

## 2. Einsatzgebiete (3)

- Beispiel 2: Flug buchen
- Passagier P und Angestellter C
  - P: Welche Flüge Hamburg – Frankfurt?
  - C: Flug LH123 um 8:30, Flug LH124 um 10:00, Flug LH125 um ...
  - P: Ich nehme LH123.
  - C: Der ist ausgebucht.
  - P: Ich nehme LH124.
  - C: Bestätigt, Buchungsnummer: 1094.

# Inhalt

1. Agententechnologie
2. Einsatzgebiete
3. Unterschiede zu OOP
4. AOP Framework
5. mental state
6. AGENT0
7. Interpreter

# 3. Unterschiede zu OOP

Framework	OOP	AOP

# 3. Unterschiede zu OOP

<b>Framework</b>	<b>OOP</b>	<b>AOP</b>
<b>Grundeinheit</b>		

# 3. Unterschiede zu OOP

<b>Framework</b>	<b>OOP</b>	<b>AOP</b>
<b>Grundeinheit</b>	Objekt	



# 3. Unterschiede zu OOP

<b>Framework</b>	<b>OOP</b>	<b>AOP</b>
<b>Grundeinheit</b>	Objekt	Agent

# 3. Unterschiede zu OOP

<b>Framework</b>	<b>OOP</b>	<b>AOP</b>
<b>Grundeinheit</b>	Objekt	Agent
<b>Parameter</b>		

# 3. Unterschiede zu OOP

<b>Framework</b>	<b>OOP</b>	<b>AOP</b>
<b>Grundeinheit</b>	Objekt	Agent
<b>Parameter</b>	frei wählbar	

# 3. Unterschiede zu OOP

<b>Framework</b>	<b>OOP</b>	<b>AOP</b>
<b>Grundeinheit</b>	Objekt	Agent
<b>Parameter</b>	frei wählbar	commitments, beliefs, ...

# 3. Unterschiede zu OOP

<b>Framework</b>	<b>OOP</b>	<b>AOP</b>
<b>Grundeinheit</b>	Objekt	Agent
<b>Parameter</b>	frei wählbar	commitments, beliefs, ...
<b>Aktionen</b>		

# 3. Unterschiede zu OOP

<b>Framework</b>	<b>OOP</b>	<b>AOP</b>
<b>Grundeinheit</b>	Objekt	Agent
<b>Parameter</b>	frei wählbar	commitments, beliefs, ...
<b>Aktionen</b>	Kommunikation	

# 3. Unterschiede zu OOP

<b>Framework</b>	<b>OOP</b>	<b>AOP</b>
<b>Grundeinheit</b>	Objekt	Agent
<b>Parameter</b>	frei wählbar	commitments, beliefs, ...
<b>Aktionen</b>	Kommunikation	Kommunikation, selbstständig

# 3. Unterschiede zu OOP

<b>Framework</b>	<b>OOP</b>	<b>AOP</b>
<b>Grundeinheit</b>	Objekt	Agent
<b>Parameter</b>	frei wählbar	commitments, beliefs, ...
<b>Aktionen</b>	Kommunikation	Kommunikation, selbstständig
<b>Nachrichten</b>		



# 3. Unterschiede zu OOP

<b>Framework</b>	<b>OOP</b>	<b>AOP</b>
<b>Grundeinheit</b>	Objekt	Agent
<b>Parameter</b>	frei wählbar	commitments, beliefs, ...
<b>Aktionen</b>	Kommunikation	Kommunikation, selbstständig
<b>Nachrichten</b>	frei wählbar	

# 3. Unterschiede zu OOP

<b>Framework</b>	<b>OOP</b>	<b>AOP</b>
<b>Grundeinheit</b>	Objekt	Agent
<b>Parameter</b>	frei wählbar	commitments, beliefs, ...
<b>Aktionen</b>	Kommunikation	Kommunikation, selbstständig
<b>Nachrichten</b>	frei wählbar	inform, request, ...

# Inhalt

1. Agententechnologie
2. Einsatzgebiete
3. Unterschiede zu OOP
4. AOP Framework
5. mental state
6. AGENT0
7. Interpreter

# 4. AOP Framework

# 4. AOP Framework

besteht aus 3 Komponenten:

# 4. AOP Framework

besteht aus 3 Komponenten:

1. formale Sprache für den mental state

# 4. AOP Framework

besteht aus 3 Komponenten:

1. formale Sprache für den mental state
2. interpretierte Programmiersprache

# 4. AOP Framework

besteht aus 3 Komponenten:

1. formale Sprache für den mental state
2. interpretierte Programmiersprache
3. Compiler von der Agentensprache zu Prozessmodellen



# Inhalt

1. Agententechnologie
2. Einsatzgebiete
3. Unterschiede zu OOP
4. AOP Framework
5. mental state
6. AGENT0
7. Interpreter

# 5. mental state

# 5. mental state

- Seelenzustand

# 5. mental state

- Seelenzustand
- es gibt keine „einzig korrekte“  
Definition

# 5. mental state

- Seelenzustand
- es gibt keine „einzig korrekte“  
Definition
- je nach Anwendungsgebiet  
erweiterbar

# 5. mental state

- Seelenzustand
- es gibt keine „einzig korrekte“  
Definition
- je nach Anwendungsgebiet  
erweiterbar
- Temporallogik

# 5. mental state

- Seelenzustand
- es gibt keine „einzig korrekte“ Definition
- je nach Anwendungsgebiet erweiterbar
- Temporallogik
  - hier explizite zeitpunktbasierte Logik

# 5. Zeitfunktion



# 5. Zeitfunktion

- Festlegung von Zeitpunkten

# 5. Zeitfunktion

- Festlegung von Zeitpunkten
- Konstante

# 5. Zeitfunktion

- Festlegung von Zeitpunkten
- Konstante
- oder Funktion

# 5. Zeitfunktion

- Festlegung von Zeitpunkten
- Konstante
- oder Funktion
- Symbol:  $t$

# 5. Beliefs

# 5. Beliefs

= Vorstellungen

# 5. Beliefs

- = Vorstellungen
- Operator B

# 5. Beliefs

= Vorstellungen

- Operator B

$B^t(a, \varphi)$



# 5. Beliefs

= Vorstellungen

- Operator B

$B^t(a, \varphi)$

$B^3(a, B^{10}(b, \text{like}(a, b)^7))$

# 5. Beliefs

=Vorstellungen

- Operator B

$B^t(a, \varphi)$

$B^3(a, B^{10}(b, \text{like}(a, b)^7))$

- Unterschied zu knowledge hier unwichtig

# 5. Commitments

# 5. Commitments

= Verpflichtungen

# 5. Commitments

- = Verpflichtungen
- Operator CMT

# 5. Commitments

= Verpflichtungen

- Operator CMT

$\text{CMT}^t(a, b, \varphi)$

# 5. Commitments

= Verpflichtungen

- Operator CMT

$\text{CMT}^t(a, b, \varphi)$

- Verbindung zu B:

# 5. Commitments

= Verpflichtungen

- Operator CMT

$$\text{CMT}^t(a, b, \varphi)$$

- Verbindung zu B:

$$\text{CMT}^t(a, b, \varphi) \equiv B^t(a, \text{CMT}(a, b, \varphi) )$$



# 5. Choice

# 5. Choice

=Auswahl

# 5. Choice

=Auswahl

- Operator CH

# 5. Choice

=Auswahl

- Operator CH

$$\text{CH}^t(\mathbf{a}, \varphi) =_{\text{def}} \text{CMT}^t(\mathbf{a}, \mathbf{a}, \varphi)$$

# 5. Choice

=Auswahl

- Operator CH

$$\text{CH}^t(\mathbf{a}, \varphi) =_{\text{def}} \text{CMT}^t(\mathbf{a}, \mathbf{a}, \varphi)$$

- Auswahl ist getroffen

# 5. Choice

=Auswahl

- Operator CH

$$\text{CH}^t(\mathbf{a}, \varphi) =_{\text{def}} \text{CMT}^t(\mathbf{a}, \mathbf{a}, \varphi)$$

- Auswahl ist getroffen

- Verpflichtung gegen sich selbst

# 5. Capabilities

# 5. Capabilities

= Befähigungen



# 5. Capabilities

- = Befähigungen
- Operator CAN

# 5. Capabilities

- = Befähigungen
- Operator CAN

$$\text{CAN}^t(a, \varphi) =_{\text{def}} \text{CH}^t(a, \varphi)$$

# 5. Capabilities

= Befähigungen

- Operator CAN

$$\text{CAN}^t(a, \varphi) =_{\text{def}} \text{CH}^t(a, \varphi)$$

- Befähigung =<sub>def</sub> Auswahl

# 5. Persistenz

# 5. Persistenz

- persistent bis Beweis Gegenteil

# 5. Persistenz

- persistent bis Beweis Gegenteil

$$B^t(a, \varphi) \wedge \neg \text{LEARN}^t(a, \neg \varphi) \supset B^{t+1}(a, \varphi)$$

# 5. Persistenz

- persistent bis Beweis Gegenteil

$$B^t(a, \varphi) \wedge \neg \text{LEARN}^t(a, \neg\varphi) \supset B^{t+1}(a, \varphi)$$

$$\neg B^t(a, \varphi) \wedge \neg \text{LEARN}^t(a, \varphi) \supset \neg B^{t+1}(a, \varphi)$$

# 5. Persistenz

- persistent bis Beweis Gegenteil

$$B^t(a, \varphi) \wedge \neg \text{LEARN}^t(a, \neg\varphi) \supset B^{t+1}(a, \varphi)$$

$$\neg B^t(a, \varphi) \wedge \neg \text{LEARN}^t(a, \varphi) \supset \neg B^{t+1}(a, \varphi)$$

$$\text{CMT}^t(a, b, \varphi) \wedge \neg \text{REVOKE}^t(a, b, \varphi) \supset \text{CMT}^{t+1}(a, b, \varphi)$$



# Inhalt

1. Agententechnologie
2. Einsatzgebiete
3. Unterschiede zu OOP
4. AOP Framework
5. mental state
6. AGENT0
7. Interpreter

# 6. AGENT0

# 6. AGENT0

- Syntax:

# 6. AGENT0

- Syntax:
  - fact statements

# 6. AGENT0

- Syntax:
  - fact statements
  - unconditional action statements

# 6. AGENT0

- Syntax:
  - fact statements
  - unconditional action statements
  - conditional action statements

# 6. AGENT0

- Syntax:
  - fact statements
  - unconditional action statements
  - conditional action statements
  - Variablen

# 6. AGENT0

- Syntax:
  - fact statements
  - unconditional action statements
  - conditional action statements
  - Variablen
  - commitment statements



# 6. AGENT0

- Syntax:
  - fact statements
  - unconditional action statements
  - conditional action statements
  - Variablen
  - commitment statements
  - Auszug aus der BNF

# 6. fact statements

# 6. fact statements

- Faktenwissen

# 6. fact statements

- Faktenwissen
- [t, fact]

# 6. fact statements

- Faktenwissen
- [t, fact]
- Beispiel:

# 6. fact statements

- Faktenwissen
- [t, fact]
- Beispiel:
  - [t, employee(smith, acme)]

# 6. unconditional actions (1)

# 6. unconditional actions (1)

- Aktionen ohne Bedingung



# 6. unconditional actions (1)

- Aktionen ohne Bedingung
- private actions

# 6. unconditional actions (1)

- Aktionen ohne Bedingung
- private actions
- können sichtbar sein

# 6. unconditional actions (1)

- Aktionen ohne Bedingung
- private actions
- können sichtbar sein
  
- $DO(t, p\text{-action})$

# 6. unconditional actions (2)

# 6. unconditional actions (2)

- communicative actions

## 6. unconditional actions (2)

- communicative actions
- INFORM(t, a, fact)

## 6. unconditional actions (2)

- communicative actions
- INFORM(*t*, *a*, *fact*)
- REQUEST(*t*, *a*, *action*)

## 6. unconditional actions (2)

- communicative actions
- INFORM(*t*, *a*, *fact*)
- REQUEST(*t*, *a*, *action*)
- UNREQUEST(*t*, *a*, *action*)



## 6. unconditional actions (2)

- communicative actions
- INFORM( $t, a, \text{fact}$ )
- REQUEST( $t, a, \text{action}$ )
- UNREQUEST( $t, a, \text{action}$ )
- REFRAIN action

# 6. conditional actions

# 6. conditional actions

- Auswertung von mental conditions

# 6. conditional actions

- Auswertung von mental conditions
- Vergleich von mental patterns

# 6. conditional actions

- Auswertung von mental conditions
- Vergleich von mental patterns
  - z.B. (B, fact)

# 6. conditional actions

- Auswertung von mental conditions
- Vergleich von mental patterns
  - z.B. (B, fact)
- Syntax:

## 6. conditional actions

- Auswertung von mental conditions
- Vergleich von mental patterns
  - z.B. (B, fact)
- Syntax:
  - IF mntlcond THEN action

## 6. conditional actions

- Auswertung von mental conditions
- Vergleich von mental patterns
  - z.B. (B, fact)
- Syntax:
  - IF mntlcond THEN action
- Beispiel:



## 6. conditional actions

- Auswertung von mental conditions
- Vergleich von mental patterns
  - z.B. (B, fact)
- Syntax:
  - IF mntlcond THEN action
- Beispiel:
  - IF (B, employee(smith, acme)) THEN

## 6. conditional actions

- Auswertung von mental conditions
- Vergleich von mental patterns
  - z.B. (B, fact)
- Syntax:
  - IF mntlcond THEN action
- Beispiel:
  - IF (B, employee(smith, acme)) THEN  
INFORM(t, a, employee(smith, acme))

# 6. Variablen

# 6. Variablen

- beginnen mit ,?`

# 6. Variablen

- beginnen mit ,?`
- ersetzen

# 6. Variablen

- beginnen mit ,?`
- ersetzen
  - Namen von Agenten

# 6. Variablen

- beginnen mit ,?`
- ersetzen
  - Namen von Agenten
  - fact statements

# 6. Variablen

- beginnen mit ,?`
- ersetzen
  - Namen von Agenten
  - fact statements
  - action statements



# 6. Variablen

- beginnen mit ,?`
- ersetzen
  - Namen von Agenten
  - fact statements
  - action statements
- Beispiel:

# 6. Variablen

- beginnen mit ,?`
- ersetzen
  - Namen von Agenten
  - fact statements
  - action statements
- Beispiel:
  - IF NOT((CMT,?x), REFRAIN action)  
THEN action

# 6. commitment statements (1)

# 6. commitment statements (1)

- bei Entwicklung unbekannte Nachrichten

# 6. commitment statements (1)

- bei Entwicklung unbekannte Nachrichten
- message pattern:

# 6. commitment statements (1)

- bei Entwicklung unbekannte Nachrichten
- message pattern:
  - (from, type, content)

# 6. commitment statements (1)

- bei Entwicklung unbekannte Nachrichten
- message pattern:
  - (from, type, content)
- Beispiel:

# 6. commitment statements (1)

- bei Entwicklung unbekannte Nachrichten
- message pattern:
  - (from, type, content)
- Beispiel:
  - (a, REQUEST, DO(t, walk) AND



# 6. commitment statements (1)

- bei Entwicklung unbekannte Nachrichten
- message pattern:
  - (from, type, content)
- Beispiel:
  - (a, REQUEST, DO(t, walk) AND
  - NOT (?x REQUEST, DO(t, chew-gum))

# 6. commitment statements (2)

# 6. commitment statements (2)

- präzise Syntax:

## 6. commitment statements (2)

- präzise Syntax:
  - COMMIT(msgcond, mntlcond, agent, action)

## 6. commitment statements (2)

- präzise Syntax:
  - COMMIT(msgcond, mntlcond, agent, action)
- Beispiel:

## 6. commitment statements (2)

- präzise Syntax:
  - COMMIT(msgcond, mntlcond, agent, action)
- Beispiel:
  - COMMIT( (?a, REQUEST, ?action),

## 6. commitment statements (2)

- präzise Syntax:
  - COMMIT(msgcond, mntlcond, agent, action)
- Beispiel:
  - COMMIT( (?a, REQUEST, ?action), (B, myfriend(?a)),

## 6. commitment statements (2)

- präzise Syntax:
  - COMMIT(msgcond, mntlcond, agent, action)
- Beispiel:
  - COMMIT( (?a, REQUEST, ?action),  
(B, myfriend(?a)),  
?a,



## 6. commitment statements (2)

- präzise Syntax:
  - COMMIT(msgcond, mntlcond, agent, action)
- Beispiel:
  - COMMIT( (?a, REQUEST, ?action),  
(B, myfriend(?a)),  
?a,  
?action)

# 6. Auszug aus der BNF

# 6. Auszug aus der BNF

$\langle \text{program} \rangle ::= \text{INITIAL BELIEFS: ( } \langle \text{fact} \rangle \text{ )}^*$

# 6. Auszug aus der BNF

$\langle \text{program} \rangle ::= \text{INITIAL BELIEFS: ( } \langle \text{fact} \rangle \text{ )}^*$

$\langle \text{fact} \rangle ::= \langle \text{time} \rangle, \text{CAN}(\langle \text{agent} \rangle, \langle \text{action} \rangle)$

# 6. Auszug aus der BNF

$\langle \text{program} \rangle ::= \text{INITIAL BELIEFS: ( } \langle \text{fact} \rangle \text{ )}^*$

$\langle \text{fact} \rangle ::= \langle \text{time} \rangle, \text{CAN}(\langle \text{agent} \rangle, \langle \text{action} \rangle)$

$\langle \text{action} \rangle ::= \text{DO}(\langle \text{time} \rangle, \langle \text{p-action} \rangle)$

# 6. Auszug aus der BNF

`<program> ::= INITIAL BELIEFS: ( <fact> )*`

`<fact> ::= <time>, CAN(<agent>,<action>)`

`<action> ::= DO( <time>, <p-action> )`

`<variable> ::= ?(alphanumericstring)`

# Inhalt

1. Agententechnologie
2. Einsatzgebiete
3. Unterschiede zu OOP
4. AOP Framework
5. mental state
6. AGENT0
7. Interpreter

# 7. Interpreter



# 7. Interpreter

- Anforderungen an Interpreter

# 7. Interpreter

- Anforderungen an Interpreter
  - Nachrichtenübermittlung

# 7. Interpreter

- Anforderungen an Interpreter
  - Nachrichtenübermittlung
  - Übersicht

# 7. Interpreter

- Anforderungen an Interpreter
  - Nachrichtenübermittlung
  - Übersicht
  - Zeitauflösung

# 7. Interpreter

- Anforderungen an Interpreter
  - Nachrichtenübermittlung
  - Übersicht
  - Zeitauflösung
  - beliefs aktualisieren

# 7. Interpreter

- Anforderungen an Interpreter
  - Nachrichtenübermittlung
  - Übersicht
  - Zeitauflösung
  - beliefs aktualisieren
  - commitments aktualisieren

# 7. Interpreter

- Anforderungen an Interpreter
  - Nachrichtenübermittlung
  - Übersicht
  - Zeitauflösung
  - beliefs aktualisieren
  - commitments aktualisieren
  - Aktionen ausführen

# 7. Interpreter

- Anforderungen an Interpreter
  - Nachrichtenübermittlung
  - Übersicht
  - Zeitauflösung
  - beliefs aktualisieren
  - commitments aktualisieren
  - Aktionen ausführen
- Nebenläufigkeit



# 7. Nachrichtenübermittlung

# 7. Nachrichtenübermittlung

- Methoden der Plattform

# 7. Nachrichtenübermittlung

- Methoden der Plattform
- Adressierung über Namen

# 7. Nachrichtenübermittlung

- Methoden der Plattform
- Adressierung über Namen
- AGENT0 bestimmt

# 7. Nachrichtenübermittlung

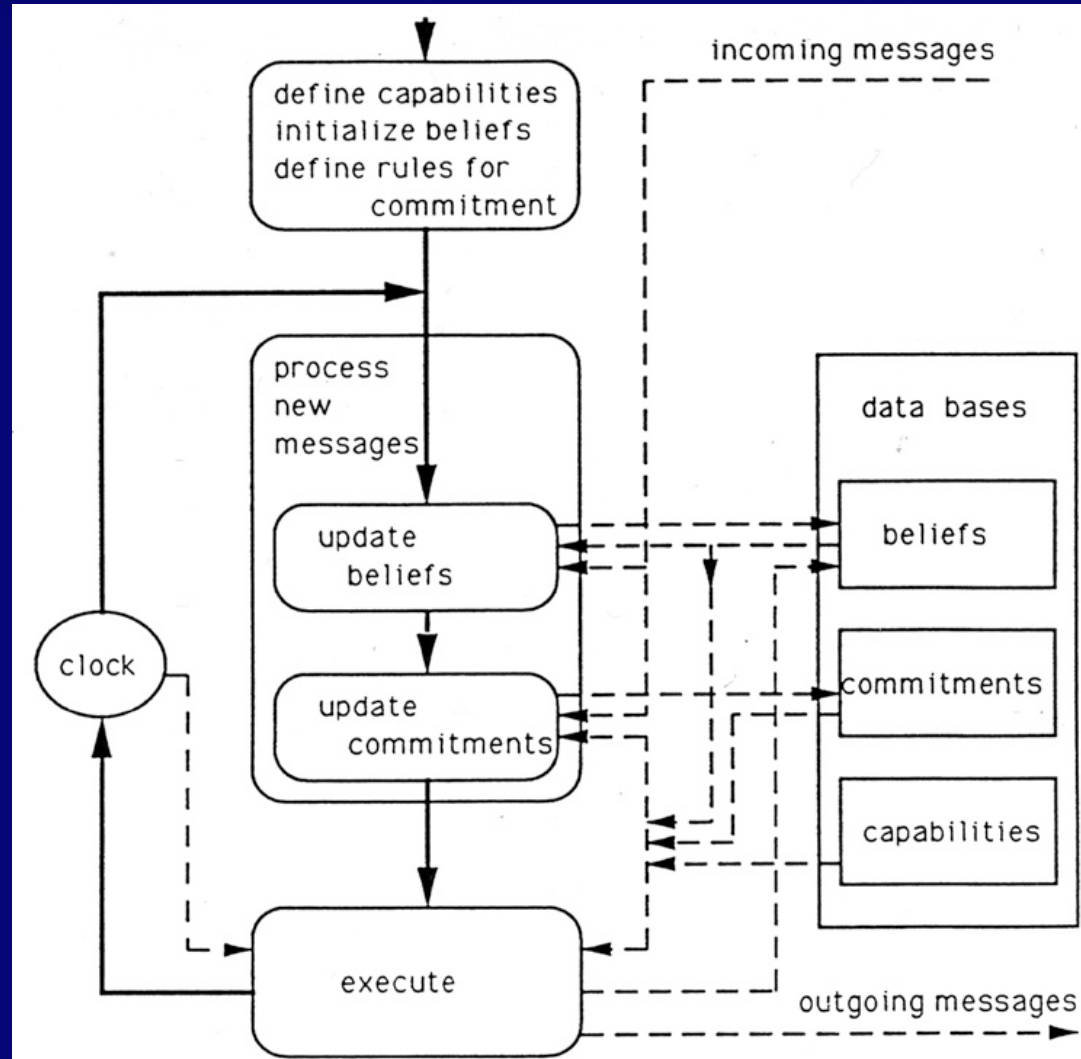
- Methoden der Plattform
- Adressierung über Namen
- AGENT0 bestimmt
  - Form

# 7. Nachrichtenübermittlung

- Methoden der Plattform
- Adressierung über Namen
- AGENT0 bestimmt
  - Form
  - Inhalt

# 7. Flussdiagramm

# 7. Flussdiagramm





# 7. Zeitauflösung

# 7. Zeitauflösung

- TIMEGRAIN: Zeitintervall

# 7. Zeitauflösung

- TIMEGRAIN: Zeitintervall
- größer als Verarbeitungszeit

# 7. Zeitauflösung

- TIMEGRAIN: Zeitintervall
- größer als Verarbeitungszeit
  - Annahme von Shoham

# 7. Zeitauflösung

- TIMEGRAIN: Zeitintervall
- größer als Verarbeitungszeit
  - Annahme von Shoham
- NOW: aktuelle Zeit

# 7. beliefs aktualisieren

# 7. beliefs aktualisieren

- 1. Schritt der Iteration

# 7. beliefs aktualisieren

- 1. Schritt der Iteration
- Agenten sind leichtgläubig



# 7. beliefs aktualisieren

- 1. Schritt der Iteration
- Agenten sind leichtgläubig
- Ablauf bei INFORM(fact) von a

# 7. beliefs aktualisieren

- 1. Schritt der Iteration
- Agenten sind leichtgläubig
- Ablauf bei INFORM(fact) von a
  - füge (a, fact) ein

# 7. beliefs aktualisieren

- 1. Schritt der Iteration
- Agenten sind leichtgläubig
- Ablauf bei INFORM(fact) von a
  - füge (a, fact) ein
  - wenn inkonsistent,

# 7. beliefs aktualisieren

- 1. Schritt der Iteration
- Agenten sind leichtgläubig
- Ablauf bei INFORM(fact) von a
  - füge (a, fact) ein
  - wenn inkonsistent,  
modifiziere alte Daten

# 7. beliefs aktualisieren

- 1. Schritt der Iteration
- Agenten sind leichtgläubig
- Ablauf bei INFORM(fact) von a
  - füge (a, fact) ein
  - wenn inkonsistent,  
modifiziere alte Daten
- Konsistenz schwer berechenbar

# 7. beliefs aktualisieren

- 1. Schritt der Iteration
- Agenten sind leichtgläubig
- Ablauf bei INFORM(fact) von a
  - füge (a, fact) ein
  - wenn inkonsistent,  
modifiziere alte Daten
- Konsistenz schwer berechenbar
- commitments unbeachtet

# 7. commitments updaten

# 7. commitments updaten

- Ablauf mit Paar (agent, action)



# 7. commitments updaten

- Ablauf mit Paar (agent, action)
  - UNREQUEST entfernt Eintrag

# 7. commitments updaten

- Ablauf mit Paar (agent, action)
  - UNREQUEST entfernt Eintrag
  - COMMIT(msgcond, mntlcond, a, action)

# 7. commitments updaten

- Ablauf mit Paar (agent, action)
  - UNREQUEST entfernt Eintrag
  - COMMIT(msgcond, mntlcond, a, action)
  - Wenn msgcond wahr ist

# 7. commitments updaten

- Ablauf mit Paar (agent, action)
  - UNREQUEST entfernt Eintrag
  - COMMIT(msgcond, mntlcond, a, action)
  - Wenn msgcond wahr ist
  - und mntlcond im mental state

# 7. commitments updaten

- Ablauf mit Paar (agent, action)
  - UNREQUEST entfernt Eintrag
  - COMMIT(msgcond, mntlcond, a, action)
  - Wenn msgcond wahr ist
  - und mntlcond im mental state
  - und Agent kann action ausführen

# 7. commitments updaten

- Ablauf mit Paar (agent, action)
  - UNREQUEST entfernt Eintrag
  - COMMIT(msgcond, mntlcond, a, action)
  - Wenn msgcond wahr ist
  - und mntlcond im mental state
  - und Agent kann action ausführen
  - kein REFRAIN action vorhanden,

# 7. commitments updaten

- Ablauf mit Paar (agent, action)
  - UNREQUEST entfernt Eintrag
  - COMMIT(msgcond, mntlcond, a, action)
  - Wenn msgcond wahr ist
  - und mntlcond im mental state
  - und Agent kann action ausführen
  - kein REFRAIN action vorhanden,
  - dann Durchführung an a melden.

# 7. Aktion ausführen (1)



# 7. Aktion ausführen (1)

- 2. Schritt der Iteration

# 7. Aktion ausführen (1)

- 2. Schritt der Iteration
- Ablauf:

# 7. Aktion ausführen (1)

- 2. Schritt der Iteration
- Ablauf:
  - überprüfen, ob action Zeitstempel hat

# 7. Aktion ausführen (1)

- 2. Schritt der Iteration
- Ablauf:
  - überprüfen, ob action Zeitstempel hat
  - ausführen, wenn

# 7. Aktion ausführen (1)

- 2. Schritt der Iteration
- Ablauf:
  - überprüfen, ob action Zeitstempel hat
  - ausführen, wenn  
Zeitstempel = NOW

# 7. Aktion ausführen (2)

# 7. Aktion ausführen (2)

- INFORM:

# 7. Aktion ausführen (2)

- INFORM:
  - Nachricht verschicken, fact einfügen



# 7. Aktion ausführen (2)

- INFORM:
  - Nachricht verschicken, fact einfügen
- REQUEST, UNREQUEST:

# 7. Aktion ausführen (2)

- INFORM:
  - Nachricht verschicken, fact einfügen
- REQUEST, UNREQUEST:
  - Nachricht versenden

# 7. Aktion ausführen (2)

- INFORM:
  - Nachricht verschicken, fact einfügen
- REQUEST, UNREQUEST:
  - Nachricht versenden
- DO:

# 7. Aktion ausführen (2)

- INFORM:
  - Nachricht verschicken, fact einfügen
- REQUEST, UNREQUEST:
  - Nachricht versenden
- DO:
  - vorher mental condition überprüfen

# 7. Aktion ausführen (2)

- INFORM:
  - Nachricht verschicken, fact einfügen
- REQUEST, UNREQUEST:
  - Nachricht versenden
- DO:
  - vorher mental condition überprüfen
- IF-THEN:

# 7. Aktion ausführen (2)

- **INFORM:**
  - Nachricht verschicken, fact einfügen
- **REQUEST, UNREQUEST:**
  - Nachricht versenden
- **DO:**
  - vorher mental condition überprüfen
- **IF-THEN:**
  - conditions überprüfen, ggf. ausführen

# 7. Nebenläufigkeit



# 7. Nebenläufigkeit

- sequentielle Iterationen





# 7. Nebenläufigkeit

- sequentielle Iterationen
- Nebenläufigkeit mehrerer Agenten



# X. Literatur

- [HYA94] Hägg, S.; Ygge, F.; An Architecture for Agent-Oriented Programming with a Programmable Model of Interaction, In Keane, Cunningham, Brady and Byrne (Eds.), Proceeding of AICS '94, pp. 115 - 128, Dublin, Ireland
- [MVT91] Mark, T.; Viola, P.; The AGENT0 Manual; 1991; Stanford University; Report STAN-CS-91-1389
- [PAR97] Parks, D.; AOP - A Practical Evaluation; 1997  
<http://www.cs.berkeley.edu/~davidp/cs263/introduction.html>
- [SHO90] Shoham, Y.; Agent-oriented programming; 1990; Stanford University; Report STAN-CS-90-1335
- [MÜL03] Müller, Dr. Jörg; Vorlesung: Multiagentensysteme; Siemens AG, CT IC 6; 2003; Teil 2: Agentenarchitekturen, Wissensrepräsentation und Reasoning

Vielen Dank für die  
Aufmerksamkeit

# 8. Beispiel (1)

# 8. Beispiel (1)

- Flugbuchung aus 2.

# 8. Beispiel (1)

- Flugbuchung aus 2.
- Makros zur Vereinfachung

# 8. Beispiel (1)

- Flugbuchung aus 2.
- Makros zur Vereinfachung
  - issue\_bp: Bordkarte zuordnen

# 8. Beispiel (1)

- Flugbuchung aus 2.
- Makros zur Vereinfachung
  - issue\_bp: Bordkarte zuordnen
  - query\_which: Infos über Flüge



# 8. Beispiel (1)

- Flugbuchung aus 2.
- Makros zur Vereinfachung
  - issue\_bp: Bordkarte zuordnen
  - query\_which: Infos über Flüge
  - query\_wether: Fluganfrage

# 8. Beispiel (1)

- Flugbuchung aus 2.
- Makros zur Vereinfachung
  - issue\_bp: Bordkarte zuordnen
  - query\_which: Infos über Flüge
  - query\_wether: Fluganfrage
- zwei COMMIT-Regeln

# 8. Beispiel (1)

- Flugbuchung aus 2.
- Makros zur Vereinfachung
  - issue\_bp: Bordkarte zuordnen
  - query\_which: Infos über Flüge
  - query\_wether: Fluganfrage
- zwei COMMIT-Regeln
  - Flug annehmen und buchen

# 8. Beispiel (1)

- Flugbuchung aus 2.
- Makros zur Vereinfachung
  - issue\_bp: Bordkarte zuordnen
  - query\_which: Infos über Flüge
  - query\_wether: Fluganfrage
- zwei COMMIT-Regeln
  - Flug annehmen und buchen
- Beispieldialog in AGENT0

# 8. Beispiel (2)

agent	action
smith	query_which(1march/1:00,smith,airline, [18april/?!time,flight(sf,ny,?!num)])
airline	INFORM(1march/2:00,smith,[18april/8:30,flight(sf,ny,#354)])
airline	INFORM(1march/2:00,smith,[18april/10:00,flight(sf,ny,#293)])
airline	INFORM(1march/2:00,smith,[18april/ ...
smith	REQUEST(1march/3:00,airline,issue_bp(smith,#354,18april))
smith	query_whether(1march/4:00,smith,airline, CMT(airline,smith,issue_bp(smith,#354,18april)))
airline	INFORM(1march/5:00,smith, NOT CMT(airline,smith,issue_bp(smith,#354,18april)))
smith	REQUEST(1march/6:00,airline,issue_bp(smith,#293,18april))
smith	query_whether(1march/7:00,smith,airline, CMT(airline,smith,issue_bp(smith,#293,18april)))
airline	INFORM(1march/8:00,smith, CMT(airline,smith,issue_bp(smith,#293,18april)))

(aus [SH090] Seite 39)