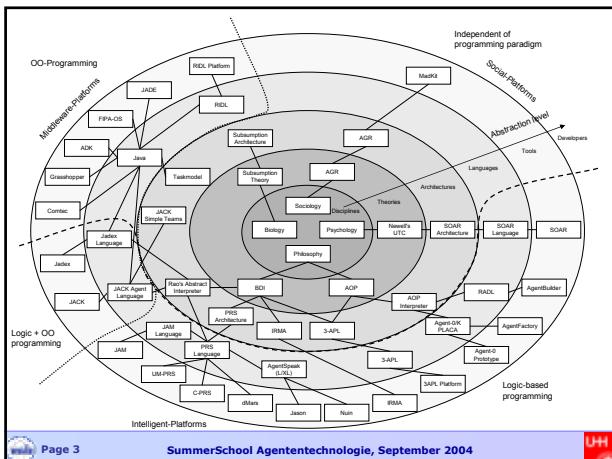


## Rational Agents

Lars Braubach, Alexander Pokahr

## What are rational agents?

- Agents that act with respect to their goals and try to achieve the best outcome
- To get rational agents the internal agent structure is the focal point of view
- -> Theories, Architectures, Languages from other disciplines



## Agent Programming Landscape (2)

### Observation

- Theoretically grounded architectures are realised as logic-based systems
- Middleware platforms are not based on theoretically grounded architectures
- Only autonomy and interaction are supported by object-oriented agent platforms (cf. weak notion of agency)
- Programming of deliberative agents is too different from mainstream OO

### Our Conclusion

- Integrate agent theories with object-orientation

## BDI (Bratman 87)

### Why BDI?

- simple!
- describes how humans think that they think (mentalistic notions)
- successful implementations already exist

### Belief, Desire, Intention -> Action

- vs. Desire-Belief-Model

### Practical Reasoning (vs. Theoretical)

- Deliberation: Selection of goals
- Means-End Reasoning: Pursuing goals
- Main concept: future-directed intention
  - Intentions are consistent in contrast to desires
  - Stability of plans (vgl. Desire-Belief-Model)

## BDI Implementation (Rao+Georgeff95)

### Beliefs, Goals, Plans

- Goals
  - conflict-free
  - modelled as events
- Beliefs and goals lead to the selection and stepwise execution of plans
- Intentions are represented implicitly by running plans

# BDI Abstract Interpreter

## BDI-interpreter

```
Initialize-state();
repeat
  options := option-generator(event-queue);
  select-options := deliberate(options);
  update-intentions(selected-options);
  execute();
  get-new-external-events();
  drop-successful-attitudes();
  drop-impossible-attitudes();
end repeat
```

Abstract BDI Interpreter [From Theory to Practice, Rao+Georgeff]



# Jadex - aktueller Stand

**Ziel:** Unterstützung der Entwicklung von rationalen Agenten mit Hilfe mentalistischer Konzepte

- Ist eine BDI-Erweiterung (add-on) der FIPA-kompatiblen JADE Multiagentenplattform
- Einfache Agentenkonstruktion durch eine XML-basierte Agentenbeschreibung und prozeduralem Planwissen in Java
- Wiederverwendbarkeit durch Agentenmodule, die Busetta's Capability Konzept folgen
- Werkzeugunterstützung durch verschiedene Toolagenten (zusätzlich zu JADE)
  - Der BDI-Viewer ermöglicht die (entfernte) Überwachung des internen Agentenzustands
  - Der BDI-Introspektor ermöglicht die (Fern)Steuerung von Agenten
  - Der Logger dient der Erfassung und justierbaren Darstellung von Log-meldungen



# Jadex - akt

**Ziel:** Unterstützung der Entwicklung von rationalen Agenten mit Hilfe mentalistischer Konzepte

- Ist eine BDI-Erweiterung der JADE Multiagentenplattform
- Einfache Agentenkonstruktion durch eine XML-basierte Agentenbeschreibung
- Wiederverwendbarkeit durch Agentenmodule, die Busetta's Capability Konzept folgen
- Werkzeugunterstützung durch verschiedene Toolagenten (zusätzlich zu JADE)
  - Der BDI-Viewer ermöglicht die (entfernte) Überwachung des internen Agentenzustands
  - Der BDI-Introspektor ermöglicht die (Fern)Steuerung von Agenten
  - Der Logger dient der Erfassung und justierbaren Darstellung von Log-meldungen



# Jadex - akt

**Ziel:** Unterstützung der Entwicklung von rationalen Agenten mit Hilfe mentalistischer Konzepte

- Ist eine BDI-Erweiterung der JADE Multiagentenplattform
- Einfache Agentenkonstruktion durch eine XML-basierte Agentenbeschreibung
- Wiederverwendbarkeit durch Agentenmodule, die Busetta's Capability Konzept folgen
- Werkzeugunterstützung durch verschiedene Toolagenten (zusätzlich zu JADE)
  - Der BDI-Viewer ermöglicht die (entfernte) Überwachung des internen Agentenzustands
  - Der BDI-Introspektor ermöglicht die (Fern)Steuerung von Agenten
  - Der Logger dient der Erfassung und justierbaren Darstellung von Log-meldungen



# Jadex - akt

**Ziel:** Unterstützung der Entwicklung von rationalen Agenten mit Hilfe mentalistischer Konzepte

- Ist eine BDI-Erweiterung der JADE Multiagentenplattform
- Einfache Agentenkonstruktion durch eine XML-basierte Agentenbeschreibung
- Wiederverwendbarkeit durch Agentenmodule, die Busetta's Capability Konzept folgen
- Werkzeugunterstützung durch verschiedene Toolagenten (zusätzlich zu JADE)
  - Der BDI-Viewer ermöglicht die (entfernte) Überwachung des internen Agentenzustands
  - Der BDI-Introspektor ermöglicht die (Fern)Steuerung von Agenten
  - Der Logger dient der Erfassung und justierbaren Darstellung von Log-meldungen



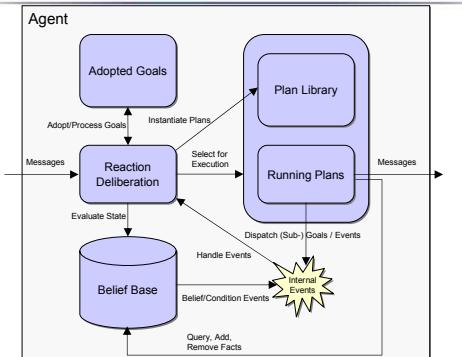
# Jadex - aktueller Stand

**Ziel:** Unterstützung der Entwicklung von rationalen Agenten mit Hilfe mentalistischer Konzepte

- Ist eine BDI-Erweiterung der JADE Multiagentenplattform
- Einfache Agentenkonstruktion durch eine XML-basierte Agentenbeschreibung
- Wiederverwendbarkeit durch Agentenmodule, die Busetta's Capability Konzept folgen
- Werkzeugunterstützung durch verschiedene Toolagenten (zusätzlich zu JADE)
  - Der BDI-Viewer ermöglicht die (entfernte) Überwachung des internen Agentenzustands
  - Der BDI-Introspektor ermöglicht die (Fern)Steuerung von Agenten
  - Der Logger dient der Erfassung und justierbaren Darstellung von Log-meldungen



## Jadex - Abstract Architecture



## Beliefbase

Repräsentation des Wissens eines Agenten

- Beliefs
- Beliefssets (Mengen)

Vorteile der Speicherung von Fakten in Beliefbase

- Erlaubt Queries auf Beliefs
- Erlaubt das Überwachen von Beliefs und Conditions -> Auslösen von Events / Goals
- Ermöglicht inter-Plan Kommunikation

## Plans

- Stellen prozedurales Wissen dar
  - Mittel zur Zielerreichung oder Reaktion auf Ereignis
  - Ähneln Jade Behaviours (stepweise Bearbeitung)
- Arten von Plans
  - Instant Plans (Servicecharakter)
    - Instantiierung bei Agentengeburt
  - Passive Plans (PRS-Style)
    - Aktivierung über Filter
- Beziehen sich auf ein Goal
  - Können Subgoals erzeugen

## Goal Description

Generic goal kinds

- achieve (a specified world state)
- query (some information)
- maintain (reestablish a specified world state whenever violated)
- perform (some action)

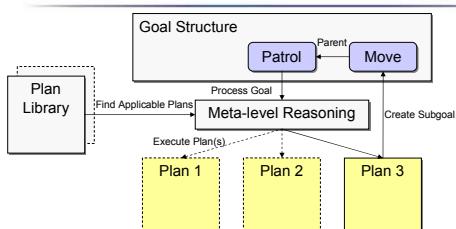
Application specific properties of goals

- Parameters represent required or optional information to be used by plans
- BDI-Flags can be used to enable non-default goal-processing

Assigning plans to goals

- Plan head indicates ability to handle goals (specified by name)
- Plan context / precondition further refines set of applicable plans

## Processing of Goals

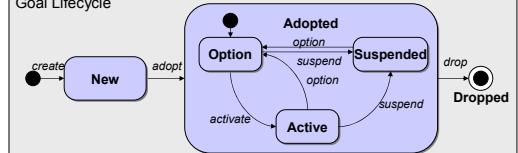


Execution cycle (BDI interpreter) - Properties to fine-tune goal behaviour

- Agents have a library of predefined (context dependent) plans
- Agents can choose among applicable plans (meta-level reasoning)
- Agents may try several plans (sequentially or at once)
- Plans may create subgoals (→ Goal Hierarchy)

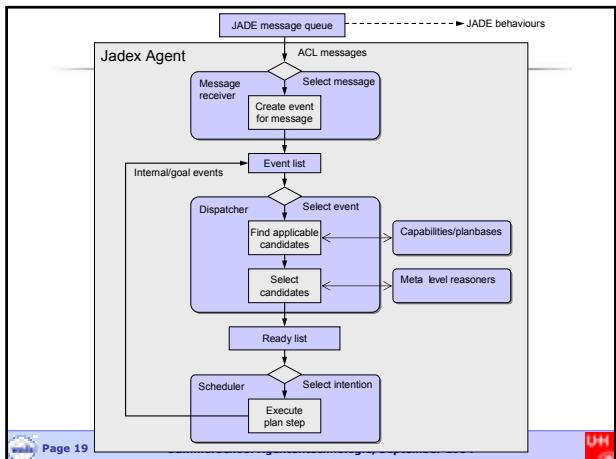
## Goal Deliberation

Goal Lifecycle



Dynamic selection of goals

- Currently has to be done manually (by the programmer)
- Some automation approaches exist (RMIT)
- Goals (and options) are known to the agent
- Pursued goals should be consistent (conflict free)
- Agents may decide which goals to pursue
- Deliberation may be based on priority, utility, conflict detection...



## Fragen

