

# Testen von Agenten in Jadex

Stefan Tittel

Universität Dortmund

Projektgruppe 491 – Wissen in Multiagentensystemen  
26. Oktober 2006

# Problemstellung und Schwierigkeiten

## Problemstellung

Im Rahmen der Qualitätssicherung sollen Agenten und ihre Komponenten automatisiert getestet werden.

## Schwierigkeiten

- kein Zugriff auf Jadex-Komponenten aus JUnit-Tests
- Jadex 0.941 stellt keine Testmöglichkeit bereit
- Testen von Agenten allgemein schwierig
  - Agentenverhalten autonom, reaktiv, proaktiv  $\Rightarrow$  kein wohldefinierter Startzustand
  - Randomisierung bei der Planselektion
  - Wechselwirkungen mit anderen Agenten oder hoher Implementierungsaufwand
  - effektives Testen von Multiagentensystemen ein offenes Problem

# Problemstellung und Schwierigkeiten

## Problemstellung

Im Rahmen der Qualitätssicherung sollen Agenten und ihre Komponenten automatisiert getestet werden.

## Schwierigkeiten

- kein Zugriff auf Jadex-Komponenten aus JUnit-Tests
- Jadex 0.941 stellt keine Testmöglichkeit bereit
- Testen von Agenten allgemein schwierig
  - Agentenverhalten autonom, reaktiv, proaktiv  $\Rightarrow$  kein wohldefinierter Startzustand
  - Randomisierung bei der Planselektion
  - Wechselwirkungen mit anderen Agenten oder hoher Implementierungsaufwand
  - effektives Testen von Multiagentensystemen ein offenes Problem

# Problemstellung und Schwierigkeiten

## Problemstellung

Im Rahmen der Qualitätssicherung sollen Agenten und ihre Komponenten automatisiert getestet werden.

## Schwierigkeiten

- kein Zugriff auf Jadex-Komponenten aus JUnit-Tests
- Jadex 0.941 stellt keine Testmöglichkeit bereit
- Testen von Agenten allgemein schwierig
  - Agentenverhalten autonom, reaktiv, proaktiv  $\Rightarrow$  kein wohldefinierter Startzustand
  - Randomisierung bei der Planselektion
  - Wechselwirkungen mit anderen Agenten oder hoher Implementierungsaufwand
  - effektives Testen von Multiagentensystemen ein offenes Problem

# Problemstellung und Schwierigkeiten

## Problemstellung

Im Rahmen der Qualitätssicherung sollen Agenten und ihre Komponenten automatisiert getestet werden.

## Schwierigkeiten

- kein Zugriff auf Jadex-Komponenten aus JUnit-Tests
- Jadex 0.941 stellt keine Testmöglichkeit bereit
- Testen von Agenten allgemein schwierig
  - Agentenverhalten autonom, reaktiv, proaktiv  $\Rightarrow$  kein wohldefinierter Startzustand
  - Randomisierung bei der Planselektion
  - Wechselwirkungen mit anderen Agenten oder hoher Implementierungsaufwand
  - effektives Testen von Multiagentensystemen ein offenes Problem

# Problemstellung und Schwierigkeiten

## Problemstellung

Im Rahmen der Qualitätssicherung sollen Agenten und ihre Komponenten automatisiert getestet werden.

## Schwierigkeiten

- kein Zugriff auf Jadex-Komponenten aus JUnit-Tests
- Jadex 0.941 stellt keine Testmöglichkeit bereit
- Testen von Agenten allgemein schwierig
  - Agentenverhalten autonom, reaktiv, proaktiv  $\Rightarrow$  kein wohldefinierter Startzustand
  - Randomisierung bei der Planselektion
  - Wechselwirkungen mit anderen Agenten oder hoher Implementierungsaufwand
  - effektives Testen von Multiagentensystemen ein offenes Problem

# Problemstellung und Schwierigkeiten

## Problemstellung

Im Rahmen der Qualitätssicherung sollen Agenten und ihre Komponenten automatisiert getestet werden.

## Schwierigkeiten

- kein Zugriff auf Jadex-Komponenten aus JUnit-Tests
- Jadex 0.941 stellt keine Testmöglichkeit bereit
- Testen von Agenten allgemein schwierig
  - Agentenverhalten autonom, reaktiv, proaktiv  $\Rightarrow$  kein wohldefinierter Startzustand
  - Randomisierung bei der Planselektion
  - Wechselwirkungen mit anderen Agenten oder hoher Implementierungsaufwand
  - effektives Testen von Multiagentensystemen ein offenes Problem

# Problemstellung und Schwierigkeiten

## Problemstellung

Im Rahmen der Qualitätssicherung sollen Agenten und ihre Komponenten automatisiert getestet werden.

## Schwierigkeiten

- kein Zugriff auf Jadex-Komponenten aus JUnit-Tests
- Jadex 0.941 stellt keine Testmöglichkeit bereit
- Testen von Agenten allgemein schwierig
  - Agentenverhalten autonom, reaktiv, proaktiv  $\Rightarrow$  kein wohldefinierter Startzustand
  - Randomisierung bei der Planselektion
  - Wechselwirkungen mit anderen Agenten oder hoher Implementierungsaufwand
  - effektives Testen von Multiagentensystemen ein offenes Problem



## Lösung: Jadex 0.95-beta1

Jadex 0.95-beta1 bietet ein an JUnit angelehntes Testcenter zum Testen einzelner Agenten.

- Jadex 0.95-beta1 ist stabil!
- Jadex 0.95 nur noch nicht veröffentlicht, weil Dokumentation noch nicht fertig
- Testfunktionalität zwar nicht dokumentiert, aber recht intuitiv
- Verwendung von Testagenten (statt Testklassen wie bei JUnit)
- `InitializeAgentPlan` kompiliert nicht ohne Anpassungen ⇒ Umstellungsaufwand

## Lösung: Jadex 0.95-beta1

Jadex 0.95-beta1 bietet ein an JUnit angelehntes Testcenter zum Testen einzelner Agenten.

- **Jadex 0.95-beta1 ist stabil!**
- Jadex 0.95 nur noch nicht veröffentlicht, weil Dokumentation noch nicht fertig
- Testfunktionalität zwar nicht dokumentiert, aber recht intuitiv
- Verwendung von Testagenten (statt Testklassen wie bei JUnit)
- `InitializeAgentPlan` kompiliert nicht ohne Anpassungen ⇒ Umstellungsaufwand

## Lösung: Jadex 0.95-beta1

Jadex 0.95-beta1 bietet ein an JUnit angelehntes Testcenter zum Testen einzelner Agenten.

- Jadex 0.95-beta1 ist stabil!
- Jadex 0.95 nur noch nicht veröffentlicht, weil Dokumentation noch nicht fertig
- Testfunktionalität zwar nicht dokumentiert, aber recht intuitiv
- Verwendung von Testagenten (statt Testklassen wie bei JUnit)
- `InitializeAgentPlan` kompiliert nicht ohne Anpassungen ⇒ Umstellungsaufwand

## Lösung: Jadex 0.95-beta1

Jadex 0.95-beta1 bietet ein an JUnit angelehntes Testcenter zum Testen einzelner Agenten.

- Jadex 0.95-beta1 ist stabil!
- Jadex 0.95 nur noch nicht veröffentlicht, weil Dokumentation noch nicht fertig
- Testfunktionalität zwar nicht dokumentiert, aber recht intuitiv
- Verwendung von Testagenten (statt Testklassen wie bei JUnit)
- `InitializeAgentPlan` kompiliert nicht ohne Anpassungen ⇒ Umstellungsaufwand

## Lösung: Jadex 0.95-beta1

Jadex 0.95-beta1 bietet ein an JUnit angelehntes Testcenter zum Testen einzelner Agenten.

- Jadex 0.95-beta1 ist stabil!
- Jadex 0.95 nur noch nicht veröffentlicht, weil Dokumentation noch nicht fertig
- Testfunktionalität zwar nicht dokumentiert, aber recht intuitiv
- Verwendung von Testagenten (statt Testklassen wie bei JUnit)
- `InitializeAgentPlan` kompiliert nicht ohne Anpassungen ⇒ Umstellungsaufwand

## Lösung: Jadex 0.95-beta1

Jadex 0.95-beta1 bietet ein an JUnit angelehntes Testcenter zum Testen einzelner Agenten.

- Jadex 0.95-beta1 ist stabil!
- Jadex 0.95 nur noch nicht veröffentlicht, weil Dokumentation noch nicht fertig
- Testfunktionalität zwar nicht dokumentiert, aber recht intuitiv
- Verwendung von Testagenten (statt Testklassen wie bei JUnit)
- `InitializeAgentPlan` kompiliert nicht ohne Anpassungen ⇒ Umstellungsaufwand

## Implementierungsbeispiel 1/4

### ADF (Auszug)

```
<capabilities>  
  <capability name="testcap" file="jadex.planlib.Test"/>  
</capabilities>  
  
...  
<beliefs>  
  <beliefsetref name="reports">  
    <concrete ref="testcap.reports"/>  
  </beliefsetref>  
  <beliefref name="testcase_cnt" class="int">  
    <concrete ref="testcap.testcase_cnt"/>  
  </beliefref>  
</beliefs>
```

## Implementierungsbeispiel 1/4

### ADF (Auszug)

```
<capabilities>  
  <capability name="testcap" file="jadex.planlib.Test"/>  
</capabilities>  
...  
<beliefs>  
  <beliefsetref name="reports">  
    <concrete ref="testcap.reports"/>  
  </beliefsetref>  
  <beliefref name="testcase_cnt" class="int">  
    <concrete ref="testcap.testcase_cnt"/>  
  </beliefref>  
</beliefs>
```



## Implementierungsbeispiel 2/4

### Fortsetzung: ADF (Auszug)

```
<plans>
  <plan name="mytestplan">
    <body>new MyTestPlan()</body>
  </plan>
</plans>

<initialstates>
  <initialstate name="default">
    <beliefs>
      <initialbelief ref="testcase_cnt">
        <fact>42</fact>
      </initialbelief>
    </beliefs>
  </initialstate>
</initialstates>
```

## Implementierungsbeispiel 2/4

### Fortsetzung: ADF (Auszug)

```
<plans>
  <plan name="mytestplan">
    <body>new MyTestPlan()</body>
  </plan>
</plans>

<initialstates>
  <initialstate name="default">
    <beliefs>
      <initialbelief ref="testcase_cnt">
        <fact>42</fact>
      </initialbelief>
    </beliefs>
  </initialstate>
</initialstates>
```

## Implementierungsbeispiel 3/4

### Fortsetzung: ADF (Auszug)

```
<plans>
  <initialplan ref="mytestplan"/>
</plans>
</initialstate>
</initialstates>
```

### MyTestPlan.java (Auszug)

```
int num = 1;
TestReport tr = new TestReport("#"+num++, "desc");
try {
  // INSERT TEST CODE HERE!
  if (someCondition == true) {
    tr.setSucceeded(true);
  }
}
```

## Implementierungsbeispiel 3/4

### Fortsetzung: ADF (Auszug)

```
<plans>  
  <initialplan ref="mytestplan"/>  
</plans>  
</initialstate>  
</initialstates>
```

### MyTestPlan.java (Auszug)

```
int num = 1;  
TestReport tr = new TestReport("#"+num++, "desc");  
try {  
  // INSERT TEST CODE HERE!  
  if (someCondition == true) {  
    tr.setSucceeded(true);  
  }  
}
```

## Implementierungsbeispiel 4/4

### Fortsetzung: MyTestPlan.java (Auszug)

```
    }  
    else {  
        tr.setReason("someCondition="+someCondition);  
    }  
    catch (someException e) {  
        tr.setReason("Failed because of "+e.toString());  
    }  
  
    getBeliefbase().getBeliefSet("reports").addFact(tr);
```

# Screenshot

The screenshot displays the Jadex Control Center 0.95-beta1 (2006/07/07) interface for an unnamed project. The window title is "Jadex Control Center 0.95-beta1 (2006/07/07): Unnamed project". The menu bar includes "File", "Model", and "Help".

The main interface is divided into several sections:

- Test suite settings:** Features a "Test cases" list box (currently empty), a "Testcase timeout [ms]:" input field set to "20000", and buttons for "Add", "Load", "Save", and "Clear".
- Test suite execution:** Shows a progress bar with the text "Performed: 1/1 in 1,6s Failed: 0/1". Below the progress bar are "Run" and "Abort" buttons.
- Details:** A scrollable area titled "Performed 1 Test Cases" showing a list of test cases. The first entry is "1 /home/tittel/workspace/testbeispiele/jadex/testcases/misc/AMSTest.agent.xml" with a green circle icon and "1,6s". Below this, a detailed log for "Testcase 1" is shown, listing 10 individual test steps, all marked with green circles, and a total execution time of "Took 1669 ms." A "Back to top" link is at the bottom of the details section.