

Netzwerke und Agenten

(1)

Rechtliche Aspekte, Einführung in POND



Mag. Dipl.-Ing. Dr. Michael Sonntag

Netzwerke und Agenten

Rechtliche Aspekte, Einführung in POND

E-Mail: sonntag@fim.uni-linz.ac.at

WWW: <http://www.fim.uni-linz.ac.at/staff/sonntag.htm>

Copyright, 2002 © Michael Sonntag



?

?

?

?

Fragen?

Bitte gleich stellen!

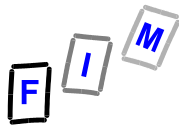
?

?

Michael Sonntag

Netzwerke und Agenten

2



Inhalt

- **Rechtliche Aspekte**
 - Rechtspersönlichkeit
 - Handlungen
- **Einführung in POND**
 - Architektur
 - Ein einfacher Agent
 - Mobilität
 - Sicherheit
 - User Interface

Michael Sonntag

Netzwerke und Agenten

3

Rechtliche Aspekte Rechtspersönlichkeit

- **Was ist ein Agent: Ist er eine Person?**
 - Natürliche Person = Mensch
 - » Wohl eher nicht!
 - Juristische Person = Was Gesetz dazu erklärt
 - » **Rechtsfähigkeit = Kann Träger von Rechten und Pflichten sein**
 - » **Handlungsfähigkeit = Kann diese durch eigenes Verhalten erwerben**
 - Kann ein Agent wirklich selbst handeln, oder handelt nicht immer ein Mensch durch ihn?
 - Hat ein Agent einen **eigenen** Willen?
 - » **Zwei Arten: Personenverbände und Sachgesamtheiten**
 - Personenverbände: Alle Arten von Gesellschaften, Vereinen, ...
 - Sachgesamtheiten: Stiftungen, Anstalten

Michael Sonntag

Netzwerke und Agenten

4



Rechtliche Aspekte Rechtspersönlichkeit

- Ist er vielleicht etwas neues?

- Was ist bei wirklich "intelligenten" Agenten?

- » Eventuell eine Analogie zu natürlichen Personen
 - » Derzeit nicht aktuell wegen mangelnder Intelligenz!

- Probleme die dabei entstehen würden:

- » In welchen Bereichen werden sie wirklich gleichbehandelt?
 - » Exekution gegen Agenten: Wie treibt man Geld ein?
 - » Deliktsfähigkeit: Wie bestraft man sie?
 - » Wie ist dann die Rolle des "Besitzers"; welches Rechtsverhältnis besteht zwischen den beiden?
 - » Was ist das Löschen eines Agenten? (Mord?)
 - » Menschenrechte ⇒ Agentenrechte?
 - »

Michael Sonntag

Netzwerke und Agenten

5



Rechtliche Aspekte Rechtspersönlichkeit

- Derzeit sicher: Agenten sind nur ein Werkzeug

- Analog zu einem Hammer oder einer Maschine

- Verantwortlich für jeden Agenten ist immer ein Mensch

- Probleme:

- Abgrenzung der Haftung Besitzer ⇔ Programmierer

- » Und dem Serverbetreiber, Netzwerksbetreiber, Verkäufer, ...

- Identifizierung des Besitzers

- » Zertifikate und digitale Signaturen

- Staatsangehörigkeit eines Agenten

- » Wenn Besitzer bekannt, kein Problem; aber sonst ...?

Michael Sonntag

Netzwerke und Agenten

6



Handlungen von Agenten Ort

- **Wo handelt ein Agent eigentlich?**
 - Wichtig für anwendbares Recht und Konsumentenschutz!
- **Agent ist immer ein Bote seines Besitzers**
 - Gleich wie wenn der Besitzer direkt handeln würde
 - » **Webbrowser,**
- **Was ist bei mobilen Agenten?**
 - Keine Tätigkeit des Partners im Besitzerstaat mehr
 - » **Siehe Konsumentenschutz sogleich!**
 - Anonyme Agenten: Ausländer-Eigenschaft für Partner nicht erkennbar
 - **Handlungsort ist, wo der Agent sich befindet**

Michael Sonntag

Netzwerke und Agenten

7



Handlungen von Agenten Konsumentenschutz

- **Konsumentenschutz stellt auf Verbraucher ab**
 - Handlungsort des Agenten daher oft egal!
 - Problem: Durchsetzbarkeit im Ausland
- **Fernabsatz: Wichtig, daß im Verbraucherstaat Werbung/Angebot erfolgte**
 - Bei mobilen Agenten nicht; dies geht daher verloren!
 - » **Unzulässige Vertragsbestandteile (§ 6 KSchG)**
 - » **Ungewöhnliche AGB-Bestimmungen (§ 864a)**
 - » **Grob benachteiligende AGB-Klauseln (§ 879 Abs 3)**
- **Fernabsatz-RL: Informationspflichten sind erfüllt, wenn die Daten dem Agenten übergeben werden**

Michael Sonntag

Netzwerke und Agenten

8



Handlungen von Agenten Erklärungsbewußtsein

- **Wichtiger Punkt von Agenten: Autonomie**
 - Besitzer kann nicht alles voraussehen
 - » Ist das ein rechtliches Problem?
 - Explizites Erklärungsbewußtsein in Ö nicht nötig
 - » Adäquate Verursachung
 - » Möglichkeit zur Vermeidung, daß Empfänger als Erklärung sieht
 - Bei Agenten:
 - » Agentenstart und Aufgabenzuweisung sind adäquat
 - » Nicht-verwenden von Agenten hätte geholfen
- **Jede Äußerung des Agenten ist voll dem Besitzer zuzurechnen; es gibt hier kein Schlupfloch!**

Michael Sonntag

Netzwerke und Agenten

9



Handlungen von Agenten Zugang von Erklärungen

- **Bedeutsam ist meist der Zugang einer Erklärung**
 - Fristablauf, Zukommen
- **Senden: Nach normalen Regeln**
- **Empfang: Agent ist Bote seines Besitzers**
 - An Agenten geliefert = Besitzer hat erhalten
 - » Wohl erst, wenn der Agent den Server verläßt, von dem er die Nachricht erhalten hat
 - » Nachricht stammt von Drittem: Sofort empfangen!
 - Unterscheidung schwierig!
 - Nachweis: Bestätigung des Agenten und Hand-off
 - » Oder: Trusted Third Party (Bestätigter Austausch)

Michael Sonntag

Netzwerke und Agenten

10



Handlungen von Agenten Anwendbares Recht

- In der Regel das vereinbarte Recht (falls erfolgt)
- Sonst: Recht des Erbringers der charakteristischen Leistung
 - Verkäufer/Dienstleister: Dessen Recht!
- Ausnahme: Konsumentenschutz (Siehe oben)
- Problem: Durchsetzung
 - Zusätzlich bei Agenten: Oft Beweisprobleme
- Daher eher: Lokale Agenten verwenden
 - Keine Beweisprobleme, Schutz bleibt erhalten

Michael Sonntag

Netzwerke und Agenten

11



POND Grundlagen


- Das Agentensystem basiert auf Java
 - Erweiterungen des Rechtesystems: Komplex, dynamisch
 - » Fallstrick bei Installation: System selbst muß alle Rechte erhalten, sonst kann es nicht starten!
 - Enthält schwache Mobilität
 - » Keine Programmierung nötig außer in Sonderfällen
 - Z. B. bei nicht-serialisierbaren Datenelementen (Fenster, ...)
 - » Hauptroutine ist Nachrichtenverarbeitungsfunktion
 - Konversationen (=Kommunikation zwischen Agenten)
 - » Konzept: Message-Passing
 - » Nur lokal
 - » Auch digital signierte Nachrichten möglich

Michael Sonntag

Netzwerke und Agenten

12






POND Architektur (1)

- **Basis-Agentensystem (Commandline)**
 - Stellt eine Art neue Betriebssystemschiicht dar
 - Bietet Dienste für Agenten: Mobilität, Kommunikation, ...
- **GUI-Teil: User-Interface für Agentensystem**
 - Agentensystem selbst und für Agenten
- **Messaging: Einfache Kommunikation**
 - Broadcasts und Unicasts
- **Experiences: Modell für verteiltes Vertrauen**
 - Hier nicht benötigt und erläutert!

Michael Sonntag

Netzwerke und Agenten

13



POND Architektur (2)

- **FIM-Utilities**
 - **Payment:** Bezahlung ist für Berechtigungen nötig
 - **SMS versenden übers Web**
 - » **Derzeit nicht funktionsfähig, da keine anonymen Dienste mehr!**
 - » **Update geplant!**
 - **Threads:** Java Threads, die unterbrochen werden können
 - » **Wichtig für das Agentensystem: Agenten beenden**
 - **WWW:** Webseiten (+Formulare) lesen und analysieren
 - **Sonstige kleinere Elemente**

Michael Sonntag

Netzwerke und Agenten

14



POND Installation (1)

- **J2EE installieren (falls nicht schon erfolgt)**
 - Java 2 Enterprise Edition (Enthält XML, ...)
 - » JRE alleine reicht NICHT aus (ist aber VORHER zu installieren)!
 - » Version 1.2 oder später ist Minimum (Getestet: 1.4)!
- **Agentensystem Release-Paket entpacken**
 - In ein lokales Verzeichnis (Bsp: "c:\java\agents")
- **Security Provider installieren**
 - Datei: .../jre/lib/security/java.security
 - Einfügen (hinten dran mit neuer Nummer statt "?"):
`security.provider.?=iaik.security.provider.IAIK`
 - » Dies ermöglicht die Verwendung der Kryptographie-Klassen

Michael Sonntag

Netzwerke und Agenten

15

POND Installation (2)

- **Security Policy setzen**
 - Datei: .../jre/lib/security/java.policy
 - `grant codeBase "file:/<<PATH>>/-"`
`{ permission java.security.AllPermission; };`
 - Statt <<PATH>> den Pfad des Agentensystem-Verzeichnisses angeben (Bsp.: "c:/java/agents/")!
 - » Dabei auf Windows-Systemen "\" durch "/" ersetzen!
 - » Dies erlaubt es dem Agentensystem, Security selbst zu managen
- **Pfade anpassen**
 - Datei: SetPath.bat (e. g. c:\java\agents\bin\SetPath.bat)
 - » In "CLASSPATH" Werte durch absolute Pfade ersetzen
 - » In "PATH" den absoluten Pfad zum JRE setzen (.../jre/bin)

Michael Sonntag

Netzwerke und Agenten

16



POND Installation (3)

- **Ausprobieren: bin\GUISystem1.bat starten**
- **Gründe für häufige Schwierigkeiten**
 - ❶ “algorithm PBE not implemented”, “algorithm ??? not found”
 - » **Security Provider Installation überprüfen!**
 - ❷ “Couldn’t establish security manager”
 - » **Security Policy überprüfen!**
 - ❸ Wie ❷ sowie generell: Überprüfen, welche Java Virtual Machine (JVM) gestartet wird (oft mehrere installiert!)
 - » **Vollständigen Pfad zu java.exe in GUISystem1.bat angeben; wenn es dann funktioniert, so ist der Pfad falsch gesetzt (Path.bat)!**
 - Unter Windows existiert oft die IE-JVM (z. B. c:\WinNt\System32)

Michael Sonntag

Netzwerke und Agenten

17



POND Installation (4)

- **Häufige Probleme beim Ausführen**
 - ➔ Werden mehrere Systeme gestartet, so wird der “keystore” (Zertifikate des Agentensystems, etc.) mehrfach eingelesen ⇒ Noch kein Problem
 - ➔ Allerdings wird er auch mehrfach geschrieben: Jedesmal wenn ein Agentensystem beendet wird. Ergebnis: Die Daten des zuerst geschlossenen Systems (z. B. neue Identitäten, Schlüssel, importierte Zertifikate) gehen verloren!

Michael Sonntag

Netzwerke und Agenten

18



POND Ein einfacher Agent (1)

- Alle Agenten sind von **AgentBase** oder **GUIAgentBase** abzuleiten

```
package Demo;
import PkgAgentSystem.*;
import PkgAgentSystem.GUI.*;

public class HelloWorld extends GUIAgentBase
{
    public HelloWorld() {}

    protected javax.swing.JPanel createVisualization()
    {
        .....
    }
}
```

Michael Sonntag

Netzwerke und Agenten

19



POND Ein einfacher Agent (2)

- Soll ein UI dargestellt werden, so muß dieses in **createVisualization** erzeugt werden

```
protected javax.swing.JPanel createVisualization()
{
    javax.swing.JPanel visualization=new javax.swing.JPanel();
    visualization.setBorder(new javax.swing.border.EtchedBorder());
    javax.swing.JLabel label=new javax.swing.JLabel("Hello, World: "+getAgentName());
    visualization.add(label);
    return visualization;
}
```

- Dieser Agent ist bereits (theoretisch) mobil und voll funktionsfähig!

Michael Sonntag

Netzwerke und Agenten

20



POND Ein einfacher Agent (3)

- Erzeugen des Agenten:
 - GUISystem1.bat starten
 - Auswählen "Agents" - "Create"
 - Eingeben: "BaseAgents.HelloWorld.HelloWorld"
= **Vollständiger Name der Hauptklasse des Agenten!**
 - "Create" - "New identity..." auswählen
 - (Beliebigen) Namen für den Agenten eingeben, "server certificate" anwählen
 - Neuen Schlüssel erzeugen ("Create new keypair..."; Namen eingeben) und auswählen
 - "OK" - Neue Identität auswählen - "Create Agent"

Michael Sonntag

Netzwerke und Agenten

21



POND Ein einfacher Agent (4)

- Vollständig mobil machen:
 - In ein Jar-Archiv einpacken und signieren
 - » **AgentHome\Makejar_HelloWorld.bat aufrufen**
 - Inhalt:

```
REM Initialisieren (Pfade, ...)  
cd ..  
call Initialize.bat  
cd AgentHome  
REM JAR-Datei erzeugen  
jar cf HelloWorld.jar BaseAgents\HelloWorld\HelloWorld.class  
REM JAR-Datei signieren  
java FIM.Util.Crypto.JarUtils HelloWorld.jar  
    ..\certificate_CAsigned.DSA "Test-Signer"
```

 - » **Die (hier alleinige) class-Datei wird in ein Archiv eingepackt**
 - » **Dieses Archiv wird signiert (Zertifikate siehe sogleich!)**

Michael Sonntag

Netzwerke und Agenten

22



Adressierung für Mobilität

- **URLs von POND-Agentensystemen sind folgendermaßen aufgebaut:**
 - Protokoll: “amp” (**A**gent **M**ovement **P**rotocol)
 - » **Proprietär!**
 - Host: Name oder IP des Zielrechners
 - Port: Port des Agentensystems
 - » **Standardwert: 4711**
 - » **Oder explizit beim Starten angeben!**
- **Beispiele:**
 - amp://AS_server.fim.uni-linz.ac.at:4711
 - amp://localhost:4712



POND Ein einfacher Agent (5)

- **Test der Mobilität (lokal):**
 - GUISystem1.bat und GUISystem2.bat starten
 - Im System 1 den Agenten erzeugen (wie vorher)
 - Rechte Maustaste auf Agenten klicken und “Move” wählen
 - Rechnernamen und Portnummer des anderen Systems eingeben (Siehe Titelzeile des Hauptfensters!)
- **Rücktransfer: Exakt gleich; nur anderer System-Name**



Zertifikate zum Test (1)

- **Kleines Hilfsprogramm erstellt Zertifikate zum Testen, sodaß keine gekauft werden müssen**
 - `java FIM.test.CreateCertificates <Ziel-Verzeichnis>`
 - Dies erfolgt automatisch!
 - » **Daher nur verwenden, wenn neue benötigt werden,**
 - Die Zertifikate gelten jeweils ein Jahr
 - » **oder weitere/an anderem Ort erstellt werden sollen!**
- **Hier werden ausschließlich DSA-signierte Zertifikate verwendet!**



Zertifikate zum Test (2)

- **Erzeugt werden folgende Dateien:**
 - `ca_certificate.dsa`: Zertifikat (selbst-signiert)
 - » **Als Zertifikate der Zertifizierungsstelle gedacht**
 - » **Sollte als "Trusted Certificate" im AS installiert werden**
 - `certificate_CAsigned.dsa`: Zertifikat, daß mit vorigem Zertifikate signiert wurde
 - » **"Normales" Benutzerzertifikat**
 - `certificate_selfsigned.dsa`: Zertifikat (selbst-signiert)
 - » **Ähnlich wie CA-Zertifikat**
 - » **Sollte aber nicht als "Trusted" eingestellt werden**
 - Zum Testen: Gültiges Zertifikate; CA unbekannt oder nicht existent



Zertifikate zum Test (3)

- **Es existiert noch ein Zertifikat, das beim ersten Start des AS automatisch erstellt wird:**
 - "Server certificate"
 - Dieses ist nicht öffentlich zugänglich (nur im Keystore)
 - Agenten mit diesem Zertifikate laufen mit den höchsten Berechtigungen (und dürfen praktisch alles)
- **Wenn es Probleme mit Berechtigungen gibt**
 - Mit diesem Zertifikat ausprobieren!



Beeper + SendBeep (1)

- **Beispiel für einfache Agenten-Kommunikation**
 - Hier nur ein einziger einfacher Broadcast
- **Beeper: GUI-Agent (Klicken: Einmal Piepsen)**
- **SendBeep: GUI-Agent (Klicken: Broadcast senden)**
- **Konversationen müssen registriert werden**
 - Im Konstruktor des Agenten einfügen:
`registerConversation(new BeepConversation(this));`
 - Ansonsten erfolgt eine Fehlermeldung/werden Nachrichten nicht empfangen!



Beeper + SendBeep (2)

- **Starten einer Konversation:**

- `Conversation conv=newConversation(
 BeepConversation.conversationTypeID,null);
 startConversation(conv);`

- 1) Konversation erzeugen

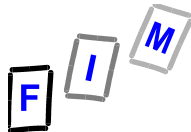
- » Benötigt die ID der Konversation und die Identität des Partner-Agenten (hier null, da es ein Broadcast ist)

- 2) Konversation starten

- » Die Nachrichtenfunktion der Konversation wird mit dem Parameter "null" ausgeführt und sollte die erste Nachricht absenden

- 3) Konversation läuft, bis sie sich selbst beendet

- » Oder: Warten auf Ende: `conv.waitEnded();`



Beeper + SendBeep (3)

- **Die Nachricht selbst:**

- Sehr kurz! Da wir keine Daten übertragen ist sie komplett leer (toString ist eigentlich nicht nötig)

```
public class BeepMessage extends BroadcastMessage
{
    public BeepMessage()
    {
    }

    public String toString()
    {
        return "Beep!";
    }
}
```



Beeper + SendBeep (4)

- **Die Konversation (=Hauptstück!):**

- Muß von “Conversation” abgeleitet werden

- Muß die ID definieren (eindeutig!):

- » `public final static long conversationTypeID=4737;`

- Wichtigstes Element: Nachrichten-Funktion

- » Wird NUR bei der Initialisierung mit “null” aufgerufen, sonst IMMER mit der empfangenen Nachricht!

- » Zu Beginn Systemnachrichten bzw. übergeordnete Klassen die Nachricht prüfen lassen:

- `if (super.handleMessage(msg)) return true;`

- » Rückgabewert: “true” falls Nachricht behandelt wurde

- Jede Conversation muß (irgendwann) enden!

- » `endConversation();`



Beeper + SendBeep (5)

- **Zur leichten Rollenverteilung: isInitiator()**

- Liefert “true” falls dieser Agent die Konversation gestartet hat; “false” falls die Konversation durch eine empfangene Nachricht erzeugt und gestartet wurde

- **Globale Variablen:**

- Neue Konversationen werden durch “clone” erzeugt:

- » Konstruktor wird nicht aufgerufen

- » Globale Variablen werden **nicht** neu initialisiert (sondern kopiert)

- Globale Variablen (z. B. Datenstrukturen) müssen bei der ersten Nachricht, Start der Konversation oder durch gesonderten Methodenauf Ruf initialisiert werden!



Beispiels-Agenten: Command-Return

- **CommandReturn**

- Enthält einige Spezialitäten, insbes. in MessengerAgent!

- » Nachricht signieren (mit eigenem Schlüssel)
- » Signatur prüfen (SendCommandConversation)
- » Zertifikat transferieren (dieses kann NICHT serialisiert werden; muß händisch erfolgen!)

- Muß unter Server-Berechtigungen laufen!

- » Sonst kein Zugriff auf Keystore möglich
 - Nötig, um Identität des Agenten auszuwählen

- Passwort für keystore: "password"



Beispiels-Agenten: Reminder

- **Reminder**

- Muß unter Server-Berechtigungen laufen!

- Enthält einen eigenen Thread

- » Nach Mobilität/Persistenz neu starten

- Sucht nach anderen Agenten (MessageBox)

- Verschickt E-Mails über weiteren Agenten

- Verschickt SMS selbst

- » Dazu werden zusätzliche Berechtigungen erworben (Mit einer Standard-Kreditkarte; keine Prüfung)
 - Benötigt: Properties, Socket
- » Derzeit nicht möglich; SMS-Dienste nicht mehr anonym zugänglich (Nötiges Update: Eingabe, Webseite bedienen)!



Beispiels-Agenten: E-Mail Sender

- **EMailSender**

- **Muß unter Server-Berechtigungen laufen!**

- » **Hier Grund ausschließlich in javax.mail**

- Zugriff auf Konfigurationsdateien sonst nicht möglich!

- **Erwirbt Berechtigungen**

- **Akzeptiert Suche nach E-Mail Sender Agent**

- » **Muß selbes Besitzer-Zertifikate haben**

- » **Muß Identität bestätigen können (Kenntnis des privaten Schlüssels aus seinem Identitäts-Zertifikat)**

- Beidseitig: Auch dieser Agent muß Kenntnis beweisen!

- » **Konversationen sind in Server- (=E-Mail-Sender) und Client- (wer E-Mails verschicken möchte) Teil getrennt**



Beispiels-Agenten: Messagebox Anzeige

- **MessageboxManagerAgent**

- **Muß unter Server-Berechtigungen laufen!**

- **Koordiniert die Anzeige von Hinweisfenstern auf versch. Rechnern**

- **Erzeugt einen Sub-Agenten für jede Nachricht**

- **Fire-and-forget Modell: Akzeptiert Aufgaben, gibt aber keine Rückmeldung über Erfolg/Mißerfolg**

- » **Hat eine Liste aller offenen Aufgaben**

- » **Mögliche Erweiterungen für Probleme:**

- Nochmals probieren

- Rückmeldung an Auftraggeber

- Andere Benachrichtigungsmethode benutzen