

## Association for Information Systems AIS Electronic Library (AISeL)

---

Wirtschaftsinformatik Proceedings 2001

Wirtschaftsinformatik

---

September 2001

# AWA - Eine Architektur eines agentenbasierten Workflow-Systems

Henrik Stormer

*Institut für Informatik, Universität Zürich, stormer@ifi.unizh.ch*

Konstantin Knorr

*Institut für Informatik, Universität Zürich, knorr@ifi.unizh.ch*

Follow this and additional works at: <http://aisel.aisnet.org/wi2001>

---

### Recommended Citation

Stormer, Henrik and Knorr, Konstantin, "AWA - Eine Architektur eines agentenbasierten Workflow-Systems" (2001).

*Wirtschaftsinformatik Proceedings 2001*. 13.

<http://aisel.aisnet.org/wi2001/13>

This material is brought to you by the Wirtschaftsinformatik at AIS Electronic Library (AISeL). It has been accepted for inclusion in Wirtschaftsinformatik Proceedings 2001 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

In: Buhl, Hans Ulrich, u.a. (Hg.) 2001. *Information Age Economy*; 5. Internationale Tagung  
Wirtschaftsinformatik 2001. Heidelberg: Physica-Verlag

ISBN: 3-7908-1427-X

© Physica-Verlag Heidelberg 2001

# AWA — Eine Architektur eines agentenbasierten Workflow-Systems

**Henrik Stormer, Konstantin Knorr**

Institut für Informatik, Universität Zürich

*Zusammenfassung: Workflow-Systeme bieten die Möglichkeit, Geschäftsprozesse zu automatisieren. Allerdings leiden heutige Systeme unter mangelnder Skalierbarkeit und unzureichender Flexibilität. Seit kurzem wird versucht, Agenten in Workflow-Umgebungen einzusetzen. Besonders Mobilität und Autonomie sind dabei erfolgversprechende Eigenschaften von Agenten. Der vorliegende Artikel gibt zunächst eine Einführung in die Workflow- und Agententechnologie und stellt dann eine neuartige Architektur eines agentenbasierten Workflow-Systems aufbauend auf vier Agententypen vor. Ein Beispiel eines Geschäftsprozesses illustriert die unterschiedlichen Agententypen.*

*Schlüsselworte: Agent, agentenbasiertes Workflow-System, Geschäftsprozess, Workflow, Workflow-Systemarchitektur*

## 1 Einleitung

Agenten und Workflow Management sind zwei Forschungsbereiche, die bisher weitgehend unabhängig voneinander untersucht wurden. Insbesondere im Bereich des E-Commerce, aber auch bei innerbetrieblichen Geschäftsprozessen, bieten Agenten Lösungsansätze für Probleme bestehender Workflow-Management-Systeme (WfMSe). Die zentralen Fragestellungen sind (1) in welchen Bereichen des Workflow-Managements Agenten erfolgversprechend eingesetzt werden kann und (2) wie in solchen Fällen die entsprechenden Lösungen aussehen. Dieser Artikel beantwortet die beiden Fragen, in dem AWA — eine Agentenbasierte Workflow-Architektur — vorgestellt und anhand eines Geschäftsprozesses einer Versicherung diskutiert wird.

Ein großflächiger Einsatz von Workflow-Systemen ist bisher nicht eingetreten. Zwei generelle Problemfelder können identifiziert werden:

- Die Einführung eines WfMSs erfordert eine umfassende Modellierung der Geschäftsprozesse, was häufig mit hohen Kosten verbunden ist. Viele Geschäftsprozesse sind nicht strukturiert genug, um durch ein Workflow-System

automatisiert zu werden. Deswegen bieten sich gut strukturierte Prozesse mit häufiger Ausführung für die Unterstützung durch ein WfMS an. Ausnahmen verursachen große Probleme, da sie bei der Modellierung nicht berücksichtigt und somit beim Ausführen des Prozesses vom System nicht abgefangen werden können.

- Zudem spielt die soziale Komponente eine wichtige Rolle. Viele Mitarbeiter haben Angst, dass durch die automatisierte Ausführung der Geschäftsprozesse ihre Arbeit zu stark kontrolliert wird. Zudem besteht die Gefahr der „Taylorisierung“: Ein Mitarbeiter muss bestimmte Aktivitäten in hoher Anzahl ausführen.

Am zweiten Punkt kann auch der Einsatz von Agenten-Technologie nicht viel ändern. Das *Exception Handling* spielt derzeit im Forschungsbereich des Workflow-Managements eine wichtige Rolle, wie z.B. die Änderung der Workflow-Spezifikation zur Laufzeit des Prozesses [Krad00, VoAl97]. Dieser Artikel geht allerdings weiterhin von einer fest vorgegeben Prozessbeschreibung aus und diskutiert eine neue Architektur für agentenbasierte Workflows (AWA). Den Schwerpunkt bilden dabei technische Fragestellungen.

Traditionelle WfMSe sind monolithisch aufgebaut, d.h., sie besitzen einen zentralen Server, der die Steuerung der ablaufenden Prozesse übernimmt [WFMC99]. In einem System mit mehreren hundert Clients benötigt man für diesen Server sehr schnelle Hard- und Software, damit das System in adäquater Geschwindigkeit abläuft. Ein Problem stellt die Ausfallsicherheit dieses Servers dar. Ein weiteres Problem stellt die Anbindung bestehender Systeme an das WfMS dar. Insbesondere in großen Unternehmen ist die Hard- und Software äußerst heterogen. Die Anbindung verschiedener Systeme erfordert die Implementierung von *Interfaces* — eine sehr zeitaufwendige Tätigkeit.

Agentenbasierte Workflow-Systeme bieten Lösungsansätze für einige dieser Probleme. Software-Agenten steuern das Workflow-System. Die grundlegende Idee von AWA ist mehrere spezialisierte Agenten zu erstellen, deren gemeinsame Aufgabe die Unterstützung der Ausführung eines Workflows ist. Hierbei wird für jede Prozessinstanz ein neuer Agent erstellt, welcher den gesamten Prozess steuert und dazu weitere spezielle Agententypen nutzt. Der Ansatz hat folgende Eigenschaften:

- Dezentrale Struktur: In Client/Server-Systemen führt der Ausfall des zentralen Servers zum Stillstand des gesamten Systems. Durch die Erstellung autonomer Prozess-Agenten, die für genau eine Prozessinstanz verantwortlich sind, entfällt der zentrale Workflow-Server, was sich positiv auf die Ausfallsicherheit auswirkt. Allgemein kann durch die dezentrale Struktur flexibler auf etwaige Probleme reagiert werden.
- Lastverteilung: Im traditionellen Workflow koordiniert eine zentrale Komponente des WfMSs die Ausführung der Workflows. Durch den Einsatz von

Agenten können die anstehenden Berechnungen auf verschiedene Komponenten aufgeteilt werden.

- Plattformunabhängigkeit: Fast alle heutigen Agentensysteme basieren auf Java und sind damit auf allen modernen Betriebssystemen lauffähig.

Der Rest des vorliegenden Artikels gliedert sich wie folgt: Grundlagen des Workflow-Managements und von Agenten beschreibt Abschnitt 2. Zusätzlich wird ein Geschäftsprozess aus dem Versicherungsumfeld diskutiert, der zur Verdeutlichung der Ausführungen im ganzen Artikel verwendet wird. Bestehende agentenbasierte Workflow-Systeme diskutiert Abschnitt 3. Abschnitt 4 stellt AWA vor. Abschließend werden in Abschnitt 5 die Ergebnisse des Artikels diskutiert und ein Ausblick gegeben.

## 2 Grundlagen

### 2.1 Workflows

Ein Workflow ist ein Geschäftsprozess, dessen Administration, Modellierung und Ausführung durch ein „aktives Softwarepaket“, genannt Workflow-System, unterstützt wird [JaBS97, LeA194]. Man unterscheidet zwischen der *Build-Time* und *Run-Time*. Zur Build-Time wird ein Geschäftsprozess in einer sog. *Workflow-Spezifikation* modelliert. Zur Run-Time werden mehrere Instanzen des Workflows nach diesem Modell erstellt. Diese Instanzen werden durch die Interaktion mit Benutzern und Programmen ausgeführt [CHRW98]. Ein WfMS besteht aus verschiedenen Komponenten, etwa zur Modellierung und Administration. Die wichtigste Komponente heißt *Workflow Engine* und ist verantwortlich für die Steuerung der unterschiedlichen Prozessinstanzen. Ein WfMS verwaltet typischerweise mehrere Prozesse und von jedem dieser Prozesse viele Instanzen.

Der Grundbaustein eines Workflows ist eine Aktivität. Alle Arbeitsschritte, die in einem Workflow durchgeführt werden müssen, lassen sich in Aktivitäten gliedern. Zur Build-Time müssen alle Aktivitäten eines Workflows erkannt und definiert werden. Dazu gehört, dass ihnen Subjekte zugeordnet werden, welche die Aktivität ausführen. Ein Subjekt kann eine natürliche Person sein aber auch ein Softwareprogramm.

Zusätzlich zu Subjekten erlauben viele Workflow-Systeme die Definition von Rollen. Eine Rolle bezeichnet einen Arbeitertyp mit einer definierten Qualifikation und Kompetenz [Sche97]. Ein Vorteil des Rollenkonzepts ist, dass sich Prozess- und Organisationsdefinition voneinander trennen lassen. Wenn ein neuer Mitarbeiter in das Unternehmen eintritt, so muss nicht die Prozessdefinition ver-

ändert werden, sondern lediglich die entsprechenden Rollen dem Mitarbeiter zugeordnet werden.

Die chronologische und logische Ordnung der Aktivitäten beschreibt der Kontroll-Fluss. Im Kontroll-Fluss ist es möglich, Bedingungen und Schleifen zu formulieren. Der Daten-Fluss beschreibt, wie die für Aktivitäten benötigten Daten zwischen den Aktivitäten weitergeleitet werden.

Zur Beschreibung von Geschäftsprozessen gibt es eine Vielzahl von Verfahren. Ereignis-gesteuerte Prozessketten (EPKs) [Sche97], State Charts [WoWe97] und Petri-Netze [Ober96] sind Beispiele.

## 2.2 Beispielprozess

Der zu untersuchende Geschäftsprozess stammt aus dem Versicherungsumfeld. Die Abwicklung eines Schadensfalles ist ein häufig anfallender und gut strukturierter Prozess, weswegen sich eine Unterstützung durch ein WfMS anbietet. Erfahrungsgemäß können 80% der Schadensfälle nach einer vorgegebenen Spezifikation bearbeitet werden, die restlichen 20% bilden Ausnahmen, die individuell bearbeitet werden müssen. Zudem ist es bei Versicherungen üblich, alle papierbasierten Unterlagen einzuscannen, so dass dem WfMS die Dokumente elektronisch zur Verfügung stehen.

Wir verwenden für die Modellierung des Prozesses eine EPK (vgl. Abbildung 1), da diese im deutschsprachigen Raum als Modellierungsverfahren verbreitet und etabliert sind. Sechsecke stellen Ereignisse, Rechtecke Funktionen und Ellipsen Rollen dar. Zwischen zwei Funktionen liegt stets ein Ereignis und zu jeder Funktion können Rollen modelliert werden, die die Subjekte festlegen [Sche97].

Abbildung 1 zeigt eine EPK für den Schadensbearbeitungsprozess einer Versicherung. Zunächst erstellt ein Sachbearbeiter eine Schadensmappe, die neben einer eindeutigen Schadensnummer, die Koordinaten des Antragstellers, die Schadensart (KfZ oder Hausrat), die geschätzte Schadenshöhe, Unfallberichte und eingereichte Rechnungen enthält (Aktivität (1) in Abbildung 1). Außerdem muss geprüft werden, ob rechtlichen Rahmenbedingungen eingehalten sind, der Schadensfall fristgerecht gemeldet wurde und die Police des Versicherungsnehmers noch gültig ist. Nach Aktivität (1) findet eine UND-Verzweigung — gekennzeichnet durch das  $\wedge$ -Symbol — zu den Aktivitäten (2) und (3)/(4) statt. In (2) erstellt ein externer Sachverständiger ein Gutachten über den angefallenen Schaden. Eine XOR-Verzweigung — dargestellt durch ein  $\times$ -Symbol — entscheidet abhängig von der Schadenshöhe zwischen (3) und (4). Bei einem geringen Schaden ( $\leq 5000$ ) vervollständigt ein Sachbearbeiter die Unterlagen zum Schadensfall und das Kundenprofil, bei einem höheren Schaden ( $> 5000$ ) erfolgt eine genauere Überprüfung der Schadenssumme.



(etwa der Abteilungsleiter oder Prokurist) auf Basis der in der Prozessinstanz erstellten Dokumente endgültig über den Schadensfall und veranlasst ggf. eine Auszahlung der geforderten Beträge (9).

Abbildung 2 zeigt einen Ausschnitt einer möglichen Organisationsdefinition. Rollen werden durch Ellipsen, Subjekte durch Rechtecke dargestellt. Es existieren acht Subjekte, die sechs unterschiedlichen Rollen zugeordnet sind. So ist Herr Schmidt beispielsweise die Rolle *Sachbearbeiter Kraftfahrzeug* zugeordnet. Über die Rollenhierarchie kann er außerdem die Rollen *Sachbearbeiter* und *Mitarbeiter* aktivieren.

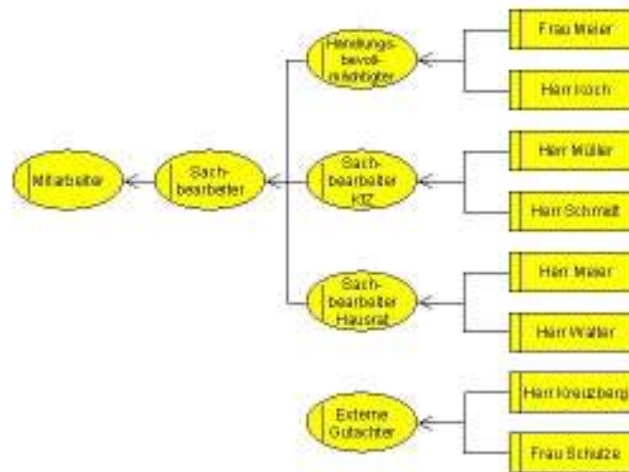


Abbildung 2: Organigramm des Beispielprozesses

Das hier betrachtete Versicherungsunternehmen bedient die Sparten KfZ und Hausrat. Die beiden Schadensabteilungen sind jedoch räumlich voneinander getrennt. Auch die externen Gutachter sind räumlich außerhalb der Versicherung angesiedelt.

### 2.3 Agenten

Der Begriff des Agenten kommt ursprünglich vom lateinischen Wort *agens*, dt. „der Treibende“. In der Informations- und Kommunikationstechnologie existiert keine einheitliche Definition. Ein (Software-)Agent unterscheidet sich nach Chorafas [Chor97] von einem „gewöhnlichen“ Programm durch folgende – in den vielen anderen Definitionen unterschiedlich stark betonten –Eigenschaften:

- **Autonom:** der Agent übt Kontrolle über seine eigenen Aktivitäten aus.



- **Adaptiv:** der Agent arbeitet zielorientiert, zum Beispiel kann er die Vorlieben des Benutzers erlernen oder sich seiner Umgebung anpassen.
- **Intelligent:** der Agent kann „intelligente“ Entscheidungen treffen, zum Beispiel durch Einbeziehung seiner Erfahrung. Im Bereich der künstlichen Intelligenz wird insbesondere die Lernfähigkeit von Softwareagenten untersucht [Ferb99].
- **Mobil:** der Agent ist in der Lage, sich selbst von einer zu einer anderen Maschine zu bewegen.

Ein mobiler Agent kann selbstständig von einem System zu einem anderen System migrieren [MiDW99] – etwa über öffentliche Netze wie das Internet. Technisch wird dabei der Quelltext des Agenten inklusive der aktuellen Variablenbelegung von einem System zum anderen geschickt. Dort angekommen, kann das Softwareprogramm wieder ausgeführt werden. Notwendige Voraussetzung für Mobilität ist ein sogenannter *Agentenplattform* auf beiden Systemen, ein Ort, auf welchem Agenten ausgeführt werden können. Die meisten Agentensysteme nutzen z.B. die Java Virtual Machine als Basis für eine Agentenplattform.

### 3 Agentenbasierte Workflow-Systeme

In einem agentenbasierten Workflow-System sind Agenten verantwortlich für die Administration, Steuerung und Koordination des Workflows. In den meisten existierenden agentenbasierten Workflow-Systemen gibt es mehrere Agententypen, die miteinander kooperieren, um einen Workflow auszuführen. Agenten können dabei auf verschiedene Weisen das Workflow-System unterstützen. Es haben sich zwei verschiedene Ansätze etabliert.

- Beim ersten Ansatz werden Agenten als kooperative Akteure eingesetzt. Wichtig an diesem Ansatz ist die Intelligenz der Agenten bei den Interaktions- und Verhandlungsstrategien. Sowohl das ADEPT Projekt [JFJB96] als auch Yu und Schmid [YuSc98] verfolgen diesen Ansatz.
- Im zweiten, in diesem Artikel genutzten Ansatz, wird für genau eine Instanz eines Workflows ein Prozess-Agent erstellt. Der Prozess-Agent muss die Prozessdefinition des Workflows lesen und damit ermitteln, welche Aktivität zu welchem Zeitpunkt von welchem Subjekt ausgeführt werden soll. Für diesen Ansatz sind die Mobilitäts- und Autonomieeigenschaften der Agenten wichtig.

Da die Anforderungen an einen solchen Prozess-Agent hoch sind, geht der Trend in der Forschung vermehrt zu spezialisierten Agenten, die den Prozess-Agenten unterstützen.

Hawryszkiewicz und Debenham teilen die Arbeit des Prozess-Agenten auf die Agententypen Control-Agent und Routing-Agent auf [HaDe98]. Ein weiterer häufig vorgeschlagener Agententyp ist der Persönliche-Agent. Er ist ortsgebunden und übernimmt für ein am Workflow beteiligtes Subjekt die Kommunikation mit anderen Agenten des Systems. Auch das DartFlow Workflow-System [CaGN96] nutzt Prozess-Agenten, die durch einen *Worklist-Server*, der die Arbeitslisten der Mitarbeiter erstellt und aktualisiert, und einen *Organization-Server*, der die Aufbauorganisation des Unternehmens beschreibt, unterstützt werden.

Merz et al. [MeLL97] untersuchen, wie Agententechnologie in zwischenbetrieblichen Workflows eingesetzt werden kann. Sie fokussieren dabei auf mobile Softwareagenten, die über eine CORBA basierte Architektur miteinander kommunizieren. Chang und Scott [ChSc96] erweitern den Persönlichen-Agent um einen Actor-Agent, der einer Instanz eines Persönlichen-Agent bezüglich einer Rolle entspricht.

Joeris [Joer00] schlägt den Einsatz eines speziellen Aktivitäts-Agenten vor, dessen Aufgabe die Ausführung genau einer Aktivität ist. Dieser Agent muss für diese Aktivität ein passendes Subjekt finden, zu diesem migrieren und nach der Erledigung der Aktivität mit den Ergebnissen zum Prozess-Agenten zurückkehren. Insbesondere die Abwicklung von parallel auszuführenden Prozessteilen ist so möglich.

Das folgende Kapitel stellt die Architektur eines agentenbasierten Workflow-Systems vor, die über die bestehenden Ansätze hinausgeht, in dem ein neuer Agententyp eingeführt wird.

## 4 AWA – eine agentenbasierte Architektur für Workflow-Systeme

AWA wurde am Institut für Informatik der Universität Zürich entwickelt [StKE00]. Sie setzt sich aus den vier Agententypen Prozess-Agent, Aktivitäts-Agent, Persönlicher-Agent und Arbeitslisten-Agent zusammen.

Ausgangspunkt ist der Prozess-Agent, welcher eine komplette Prozess-Instanz steuert. Er erstellt dazu für jede Aktivität zum richtigen Zeitpunkt einen Aktivitäts-Agent. Ein Aktivitäts-Agent ist verantwortlich für genau eine Aktivität. Der Prozess-Agent übergibt ihm alle notwendigen Daten sowie einer Auflistung, der der Aktivität zugeordneten Rollen. Daraufhin ist es seine Aufgabe, ein Subjekt zu finden, das die Aktivität ausführen kann. AWA nutzt das Rollenkonzept, in dem mindestens eine Rolle an eine Aktivität gebunden wird. Subjekten können dann Rollen zugeordnet werden. Damit der Aktivitäts-Agent ein passendes Subjekt für seine Rolle findet, kontaktiert er einen Arbeitslisten-Agent. Der Arbeitslisten-

Agent ist die wesentliche Neuerung gegenüber vorhandenen Architekturen. Er speichert eine Zuordnung der Subjekte zu den entsprechenden Rollen. Von ihm bekommt der Aktivitäts-Agent ein Subjekt zugeteilt. Anschließend migriert der Aktivitäts-Agent zum Persönlichen-Agent des Subjekts, dem er mitteilt, dass eine neue Aktivität zu erledigen ist. Der Persönliche-Agent teilt dies dem Subjekt mit und übergibt außerdem die zur Bearbeitung notwendigen Daten vom Aktivitäts-Agent. Nachdem das Subjekt die Aktivität bearbeitet hat, migriert der Aktivitäts-Agent zurück zum Prozess-Agenten und übergibt die veränderten Daten. Der Lebenszyklus eines Aktivitäts-Agenten besteht aus vier Phasen und beschreibt das Zusammenspiel der verschiedenen Agententypen. In Abbildung 3 sind diese vier Schritte aufgezeigt. Sie werden im folgenden näher erläutert.

### **Erstellen des Aktivitäts-Agent (Schritt 1)**

Der Prozess-Agent muss in der Lage sein, die nächste zu bearbeitende Aktivität zu ermitteln, wobei Vorbedingungen und Schleifen zu berücksichtigen sind. Außerdem sollte er feststellen können, ob mehrere Aktivitäten parallel ausgeführt werden können. Hat er alle zu bearbeitenden Aktivitäten ermittelt, kann für jede Aktivität genau ein Aktivitäts-Agent erstellt werden.

Dieser bekommt die notwendigen Daten sowie die der Aktivität zugeordnete(n) Rolle(n) vom Prozess-Agent übergeben und sucht dann nach einem passenden Subjekt.

Die vorgestellte Architektur soll anhand des obigen Beispiels verdeutlicht werden: Herr Weber meldet einen KfZ-Unfallsschaden mit einer Schadenssumme größer 5000 Geldeinheiten. Der Schadensmeldung wird die eindeutige Bearbeitungsnummer *251-Weber* zugeordnet, welche auch als Identifikationsnummer der Prozessinstanz dient. Der Prozess-Agent der Prozessinstanz *251-Weber* erzeugt einen Aktivitäts-Agenten für die unterschiedlichen Aktivitäten. Der Prozess-Agent erstellt beispielsweise für die Aktivität (7) *Validiere Unfallschaden* einen Aktivitäts-Agenten, dessen Lebenszyklus nun exemplarisch diskutiert wird.

### **Beim Arbeitslisten-Agent (Schritt 2)**

Der Aktivitäts-Agent sucht den Arbeitslisten-Agent auf. Der Arbeitslisten-Agent besitzt eine Liste mit allen Zuordnungen zwischen Rollen und Subjekten, also die komplette Rollendefinition des Prozesses. Sobald sich an der Rollendefinition etwas ändert, also beispielsweise ein Subjekt hinzukommt, müssen die Arbeitslisten-Agenten informiert werden.

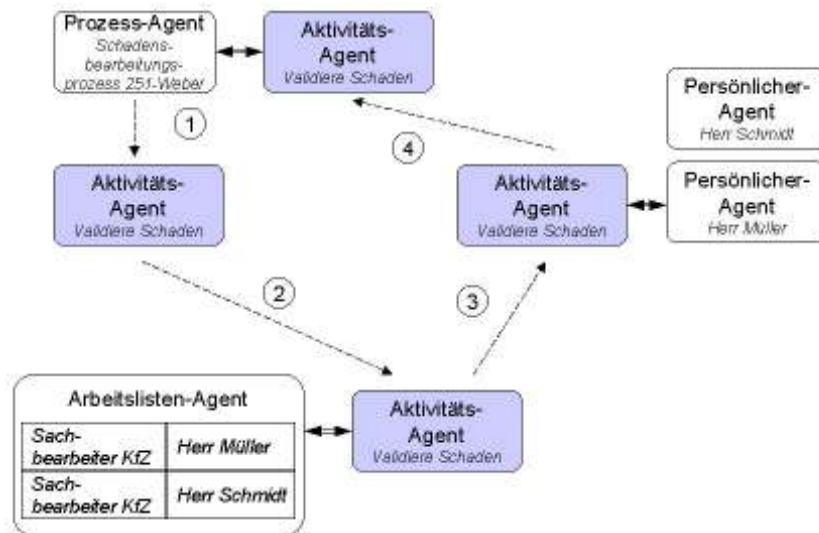


Abbildung 3: Lebenszyklus eines Aktivitäts-Agenten

Der Arbeitslisten-Agent bekommt vom Aktivitäts-Agent eine Liste mit allen Rollen dieser Aktivität übergeben und entscheidet auf Grund der Rollen und einer zusätzlichen *Policy*, welches Subjekt für die Aktivität ausgewählt wird. Beispiele für eine *Policy* sind:

1. **Kosten:** Jedes Subjekt assoziiert mit der Ausführung einer Aktivität Kosten. Der Aktivitäts-Agent könnte dann das Subjekt mit den niedrigsten Kosten aufsuchen. Auch marktbasierende Verfahren sind denkbar [HaLy98, GeKT98].
2. **Arbeitslast:** Um alle Personen, die für eine Aktivität in Frage kommen, gleich stark zu belasten, kann die Anzahl der bereits ausgeführten Aktivitäten als Basis der Entscheidung dienen.
3. **Sicherheitsaspekte:** Ein Sachbearbeiter der Versicherung sollte nicht seinen eigenen Schadensfall bearbeiten. Um Betrug zu verhindern, können bestimmte Regeln aufgestellt werden (sog. *Separation of Duties*-Regeln), die bei der Auswahl des Subjekts berücksichtigt werden müssen. Kommen mehrere Subjekte in Frage, kann eine zufällige Vergabe Betrug erschweren [StKE00].

4. Freiwillige Meldung eines Subjekts: Viele Workflow-Systeme vertrauen darauf, dass sich Subjekte freiwillig für eine anstehende Aktivität melden [Stor00].
5. Intelligentes Auswählen aufgrund von zusätzlichen Informationen: Wenn der Arbeitslisten-Agent auf weitere Informationsquellen Zugriff hat, bspw. auf die Terminkalender oder persönliche Präferenzen der Subjekte, so könnte er diese Informationen in die Auswahl des Subjektes einfließen lassen. Ein Mitarbeiter im Urlaub würde auf diese Weise keine Aktivitäten zugewiesen bekommen.
6. eine Kombination dieser Kriterien

Die Zuordnung einer Aktivität zu einem Subjekt über Rollen wird in den anderen agentenbasierten Architekturen meist vom Prozess-Agenten vorgenommen. Die Ausgliederung in einen eigenen Agententypen hat mehrere Vorteile.

- Der Arbeitslisten-Agent kann zusätzlich Aufgaben wie Monitoring oder Protokollierung übernehmen, die in agentenbasierten Workflow-Systemen schwierig zu realisieren sind.
- Bei einer Veränderung der Rollen/Subjekt-Zuweisung kann dynamisch vorgenommen werden. Viele bestehende Workflow-Systeme erlauben dies nur bei neu zu erstellenden Prozessinstanzen
- In die Entscheidung des Arbeitslisten-Agenten können auch Erfahrungen aus bereits beendeten Prozessinstanzen miteinfließen.

Im Beispiel kontaktiert der Aktivitäts-Agent den Arbeitslisten-Agent und übergibt diesem die Rolle *Mitarbeiter Kraftfahrzeug*. Hat Herr Schmidt in der Vergangenheit mehr Antragsprüfungen als Herr Müller vorgenommen, so könnte der Arbeitslisten-Agent auf Grund des zweiten Kriteriums Herrn Müller für die Prüfung dieses Schadensfalles auswählen.

### Migrieren zum Subjekt (Schritt 3)

Wenn der Arbeitslisten-Agent das Subjekt erhalten hat, migriert er zu diesem und kontaktiert dort den Persönlichen-Agenten. Ein Vorteil des Persönlichen-Agenten ist, dass es sehr einfach ist, verschiedene Schnittstellen zum Subjekt bereitzustellen, insbesondere wenn das Subjekt keine natürliche Person ist. Ein weiterer Vorteil ist die Erweiterbarkeit des Persönlichen-Agenten. Denkbar ist eine Vorfiltrierung der auszuführenden Aktivitäten für das Subjekt oder eine Weiterleitung der Aktivität bei Problemen. Evtl. können einige Aktivitäten vom Persönlichen-Agenten auch komplett oder zumindest teilweise bearbeitet werden.

Im Beispiel migriert der Aktivitäts-Agent zu Herrn Müllers Persönlichem-Agenten und übergibt diesem die Aufgabe sowie die benötigten Daten, in diesem Falle die

Informationen zum Unfall von Herrn Weber. Damit kann Herr Müller den Antrag validieren.

### **Zurück zum Prozess-Agent (Schritt 4)**

Nachdem das Subjekt die Aktivität bearbeitet hat, migriert der Aktivitäts-Agent zurück zum Prozess-Agenten und übergibt die Daten. Danach kann der Aktivitäts-Agent gelöscht werden. Sinnvoll ist zuvor eine „Archivierung“ des Agenten.

Im Beispiel migriert der Aktivitäts-Agent zum Prozess-Agent zurück und übergibt dort das Ergebnis der Validierung Herrn Müller. Danach muss die nächste auszuführende Aktivität bestimmt werden.

## **5 Diskussion und Ausblick**

Agenten eignen sich besonders für den Einsatz in zwischenbetrieblichen Workflows, also Prozessen, die mehrere Unternehmen verbinden, da sie die häufig vorhandenen heterogenen Plattformen überbrücken können. Der Beispielprozess ist ein zwischenbetrieblicher Workflow, da ein externer Gutachter in den Prozess eingebunden ist. Zusätzlich sind die beiden Schadensabteilungen räumlich voneinander getrennt.

Weiterhin unterstützt AWA den Einsatz von mobilen Geräten, wie Mobiltelefone oder persönliche digitale Assistenten. Der Aktivitäts-Agent migriert mit allen notwendigen Daten auf ein mobiles Gerät. Das Subjekt kann danach die Netzwerkverbindung unterbrechen und die Aktivität ausführen. Der Agent wartet nach dem Beenden der Aktivität so lange, bis eine Netzwerkverbindung zum Prozess-Agenten wieder aufgebaut wird und migriert dann zurück. In unserem Beispiel könnte ein externer Gutachter die notwendigen Dokumente direkt auf seinen Laptop laden und dann am Ort des Unfallgeschehens das Gutachten erstellen.

AWA wurde unter anderem entwickelt, um rollenbasierte Zugriffskontrolle und *Separation of Duties* in Workflow-Systemen zu ermöglichen. Der Arbeitslisten-Agent kann in seiner Funktionalität erweitert werden, um eine Filterung der Subjekte vorzunehmen. So können geschäftsprozess-bezogene Sicherheitsregeln durchgesetzt werden, um beispielsweise zu verhindern, dass ein Sachbearbeiter über seinen eigenen Schadensfall befindet [StKE00].

Diesen Vorteilen stehen einige Nachteile der AWA gegenüber. So ist die Entwicklung von Agentensystemen im Gegensatz zu Client/Server-Systemen noch nicht sehr weit fortgeschritten. Es existieren zwar mittlerweile zahlreiche Publikationen und Prototypen, in der Praxis werden solche Systeme zur Zeit kaum genutzt. Zudem ist die Sicherheit der Agenten ein Problem [Plei99]. Wenn ein Agent

auf ein entferntes System migriert, so ist er diesem ausgeliefert. Das System kann die privaten Daten des Agenten auslesen oder ihn einfach nicht ausführen.

Mit dem Grasshopper Agenten System wurde ein Prototyp in Java entwickelt, welche die beschriebene Architektur verifiziert. Bei der Entwicklung des Prototyps standen neben der Unterstützung von mobilen Endgeräten Sicherheitsfragestellungen im Vordergrund.

Wir erwarten, dass sich die Agentensysteme in den nächsten Jahren weiterentwickeln und so die heutigen Nachteile dieser Technologie nicht mehr so stark ins Gewicht fallen werden. Allerdings können auch Agenten nicht alle Probleme des Workflow-Managements lösen. Agentenbasierte Workflow-Systeme stellen aber eine interessante Alternative für die Zukunft dar. Insbesondere im Rahmen des E-Commerce sehen wir gute Zukunftsperspektiven für Workflows und Agenten.

## Literatur

- [CaGN96] *Cai, T.; Gloor, P.A.; Nog, S.*: DartFlow: A Workflow Management System on the Web using Transportable Agents, Technical Report PCS-TR96-283, Dartmouth College, 1996
- [Chor97] *Chorafas, D.N.*: Agent Technology Handbook, McGraw-Hill Books, 1997
- [CHRW98] *Cichocki, A.; Helal, A.; Rusinkiewicz, M.; Woelk, D.*: Workflow and Process Automation - Concepts and Technology, Kluwer Academic, 1998
- [ChSc96] *Chang, J.W.; Scott, C.T.*: Agent-based Workflow: TRP Support Environment (TSE), In: Computer Networks and ISDN Systems, 1996, S. 1501ff
- [Ferb99] *Ferber, J.*: Multi-Agent Systems: An Introduction to Artificial Intelligence, Addison Wesley Publishing Company, 1999
- [GeKT98] *Geppert, A.; Kradolfer, M.; Tombros, D.*: Market-Based Workflow Management, In: International Journal of Cooperative Information Science, 1998, Vol. 7, Nr. 4, S. 297-314
- [HaDe98] *Hawryszkiewicz, I.; Debenham, J.*: A Workflow System Based on Agents, In: Proceedings of the 1998 Database and Expert Systems Applications (DEXA) Conference, S. 135-144
- [HaLy98] *Harker, P.T.; Lyle, H.U.*: A Market-Based Approach to Workflow Automation, Department of Systems Engineering and Computer and Information Science; University of Pennsylvania, <http://www.cis.upenn.edu/~ungar/NSF.html>
- [JaBS97] *Jablonski, S.; Böhm, M.; Schulze, W.*: Workflow-Management - Entwicklung von Anwendungen und Systemen - Facetten einer neuen Technologie, dpunkt.verlag, 1997

- [JFJB96] Jennings, N.R.; Faratin, P.; Johnson, M.J.; Brien, P.O.; Wiegand M.E.: Using Intelligent Agents to Manage Business Processes, In: Proceedings of First International Conference on The Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM96), S. 345-360
- [Joer00] Joeris, G.: Decentralized and Flexible Workflow Enactment Based on Task Coordination Agents, In: Proceedings of the Conference Workshop on Agent-Oriented Information Systems, AOIS, 2000
- [Krad00] Kradolfer, M.: A Workflow Metamodel Supporting Dynamic, Reuse-based Model Evolution, Dissertation, Institut für Informatik, Universität Zürich, 2000
- [LeAl94] Leymann, F.; Altenhuber, W.: Managing Business Processes as an Information Resource, IBM Systems Journal, 1994, Vol. 33, Nr. 2, S. 326-348
- [LaOs99] Lange, D.B.; Oshima, M.: Seven Good Reasons for Mobile Agents, Communications of the ACM, 1999, Vol. 42, Nr. 3, S. 88ff
- [MiDW99] Milojicic, D.; Douglis, F.; Wheeler, R. (Editoren): Mobility – Processes, Computers, and Agents, Addison Wesley Longman Inc., 1999
- [MeLL97] Merz, M.; Liberman, B.; Lamersdorf, W.: Using Mobile Agents to support Interorganizational Workflow Management, International Journal on Applied Artificial Intelligence, 1997, Nr. 11, September, S. 551ff
- [Ober96] Oberweis, A.: Modellierung und Ausführung von Workflows mit Petri-Netzen, Teubner-Reihe Wirtschaftsinformatik, B.G. Teubner, 1996
- [Plei99] Pleisch, S.: State of the Art of Mobile Agent Computing – Security, Fault Tolerance, and Transaction Support, Research Report, IBM Research, 1999
- [Sche97] Scheer, A.W.: Wirtschaftsinformatik: Referenzmodelle für industrielle Geschäftsprozesse, Springer Verlag, Berlin, 1997
- [StKE00] Stormer, H.; Knorr, K.; Eloff, J.: A Model for Security in Agent-based Workflows, Informatik/Informatique, Zeitschrift der schweizerischen Informatikorganisationen, Nr. 6, Dezember 2000
- [Stor00] Stormer, H.: Task Distribution in Agent-Based Workflows, In: Proceedings of the ICSC Symposium on Multi-Agents and Mobile Agents in Virtual Organizations and E-Commerce (MAMA) 2000, Wollongong, Australia, Dec. 2000
- [VoAl97] Voorhoeve, M.; van der Aalst, W.: Ad-hoc Workflow: Problems and Solutions, In: Proceedings of the 8th International Workshop on Database and Expert Systems Applications (DEXA), 1997, S. 36-40
- [WFMC99] Workflow Management Coalition: Terminology and Glossary, Document Number WFMC-TC-1011, 1999, <http://www.aiim.org/wfmc/>
- [WoWe97] Wodtke, D.; Weikum, G.: A Formal Foundation for Distributed Workflow Execution Based on State Charts, In: Proc. of the 6th International Conference on Database Theory - ICDT, Delphi, Greece, January 8-10, 1997, pp. 230-246
- [YuSc98] Yu, L.; Schmid, B.: A Conceptual Framework for Agent Oriented and Role Based Workflow Modelling. In: Computer Networks and ISDN Systems, 1998