

# Konsistenzsicherung in agentenbasierten Informationssystemen

Ludger Zachewitz

Oldenburger Forschungs- und Entwicklungsinstitut für  
Informatik-Werkzeuge und -Systeme (OFFIS)  
zachewitz@offis.de

## Zusammenfassung

Durch den Einsatz von Agenten können Informationssysteme (IS) mit proaktiven und autonomen Eigenschaften realisiert werden. Jeder in einem solchen agentenbasierten Informationssystem agierende Agent besitzt Informationen, die er zur Erreichung seiner Ziele verwendet. Die Informationen eines Agenten können sich im Laufe der Zeit z. B. durch Aktualisierungen ändern und mit den Informationen anderer Agenten differieren. Schlimmstenfalls widersprechen sich Informationen von Agenten, so dass ein inkonsistenter Zustand für das Gesamtsystem zu beobachten ist. Für Agentensysteme existieren bislang keine Ansätze, die dieses Problem befriedigend lösen. In diesem Beitrag wird der Einsatz einer Wartungskomponente zur Wiederherstellung und Erhaltung konsistenter Zustände für Agentensysteme diskutiert. Dabei werden von der Wartungskomponente zu berücksichtigende Probleme/Aspekte abgeleitet. Unter Bezugnahme auf die Foundation for Intelligent Physical Agents (FIPA) wird weiterhin gezeigt, wie eine Wartungskomponente in das FIPA-Referenz-Modell für Agentensysteme einbettet werden kann.

Der Beitrag gliedert sich wie folgt: Nach der Einleitung und Motivation werden Grundlagen zu Multiagentensystemen (MAS) sowie der von der FIPA definierte Standard für MAS vorgestellt. Im dritten Kapitel wird auf das mit Agenten inhärente Problem von Inkonsistenzen und der Wartung als Methode zur Sicherung und Wiederherstellung der Konsistenz eingegangen. Ausgewählte Wartungsdimensionen werden vorgestellt, bevor im anschließenden Abschnitt eine Wartungskomponente vorgestellt wird, die das FIPA-Referenzmodell erweitert. Der Beitrag schließt mit einer Zusammenfassung und einem Ausblick.

## 1 Einleitung

Informationssysteme (IS) müssen Daten effizient verwalten und Anwendern zur Verfügung stellen. Darüber hinaus sollen heute bspw. auch Datenanalysen automatisiert erfolgen, um neues Wissen über den Datenbestand abzuleiten, oder neue, externe Datenquellen sollen identifiziert und (ad hoc) ins System integriert werden. Anwender sollen bei Auswertungen gezielt durch individuelle Hinweise oder Empfehlungen unterstützt werden, indem z. B. Analysen der Anwender-Profile (die u. a. Auskunft über den Anwender interessierende Informationen geben können) erfolgen. Komplexe IS-Strukturen und Prozesse sollten dabei für Nutzer möglichst verborgen bleiben und die System-Administration sollte einfach möglich sein. IS mit rein reaktiven Eigenschaften, die also nur auf Anforderungen oder Anweisungen von Nutzern reagieren, reichen hier nicht aus. Zur Realisierung von IS mit den genannten Fähigkeiten ist insbesondere der Einsatz autonomer und proaktiver Komponenten sinnvoll.

In diesem Zusammenhang bietet sich die Agentenmetapher als Ausgangspunkt zur Diskussion geeigneter IS-Architekturen an, um agentenbasierte Informationssysteme mit den verlangten Fähigkeiten zu entwerfen; also Systeme, die auf Veränderungen entsprechend reagieren, die selbständig in Hinblick auf die Erfüllung definierter Ziele Aufgaben initiieren und die ohne direktes Eingreifen von Anwendern Aufgaben erfüllen.

Neben den Vorteilen, die ein Anwender durch ein auf einer Agenten-Metapher basierendes System erhält, treten gesonderte Probleme mit dem Einsatz von Agenten auf. Die Ursache dieser Probleme, insbesondere des Problems der Sicherung der Konsistenz, ist in der mit

Agenten inhärenten Verteiltheit von Architektur und Information zu finden. Das Problem und entsprechende Vorschläge um es zu lösen, werden in Kapitel 3 präsentiert.

## 2 Multiagenten Systeme

Bevor in Kapitel 3 auf das Problem der Konsistenzsicherung in Agentensystemen eingegangen wird, werden nun in diesem Abschnitt Grundlagen zu Multiagentensystemen sowie des von der FIPA spezifizierten Referenz-Modells für MAS vorgestellt.

### 2.1 Grundlagen

Ein (Software-)Agent ist ein softwarebasiertes Computersystem, welches mindestens die vier Eigenschaften *Autonomie*, *Reaktivität*, *Proaktivität* sowie *soziale Fähigkeiten* aufweist [5]. *Autonomie* bedeutet in diesem Zusammenhang, dass Agenten ohne direktes Eingreifen von Benutzern operieren können und die vollständige Kontrolle über ihren inneren Zustand und über ihre Aktionen besitzen. Agenten agieren dabei stets in einer Umgebung. Sind sie zur Wahrnehmung der Umgebung sowie der Veränderungen an ihr fähig und reagieren entsprechend darauf, so besitzen sie die Eigenschaft der *Reaktivität*. Im direkten Kontrast zur Eigenschaft der Reaktivität steht die der *Proaktivität*. Sie bezeichnet die Fähigkeit eines Agenten nicht nur auf Veränderungen seiner Umgebung reagieren, sondern auch selbst Handlungen zur Erfüllung seines Zieles initiieren zu können. Proaktivität weist also auf ein zielorientiertes Verhalten hin, bei dem die Agenten selbst die Initiative ergreifen können. Die Eigenschaft eines Agenten zum *sozialen Verhalten* bezeichnet die Fähigkeit zur Kommunikation mit anderen Agenten oder mit Anwendern. Die Kommunikation erfolgt in einer Agentensprache (engl.: agent communication language (ACL)) und kann zentral (z. B. Blackboard) oder dezentral (z. B. durch direkte Agent-zu-Agent-Kommunikation) erfolgen. Sie dient der Koordinierung und der Kooperation zur Erfüllung des vom jeweiligen Agenten verfolgten Zieles.

Neben diesen zwingend erforderlichen Eigenschaften können Agenten optional die Eigenschaft der *Mobilität* aufweisen. Ein mobiler Agent hat die Fähigkeit, sich in einem Netzwerk von Computersystemen zu bewegen und auf ausgewählte Plattformen zu *migrieren*.

Operieren mehrere Agenten in einer Umgebung bzw. auf einer Plattform, spricht man von einem *Multiagentensystem* (MAS). Durch ein MAS werden Kommunikations- und Interaktionskanäle sowie -protokolle spezifiziert, d. h. es wird festgelegt in welcher Sprache kommuniziert wird, welche (z. B. Handshaking-)Verfahren dabei angewandt werden und ob z. B. eine zentrale oder dezentrale Kommunikation erfolgt.

### 2.2 Standards

Um die Entwicklung heterogener MAS zu vermeiden wurden Standards zu deren Architektur durch verschiedene Organisationen festgelegt. Nach bspw. dem von der *Foundation for Intelligent Physical Agents* (FIPA) erlassenen Standard besteht ein MAS aus mehreren Agenten, die jeweils Dienste bereit stellen können. In jedem MAS existiert genau ein *Agent Management System* (AMS) bei dem sich Agenten, die auf die Plattform migrieren möchten, registrieren müssen. Da jeder auf der Plattform befindliche Agent sich beim AMS registriert, besitzt das AMS die Fähigkeit, einen „White-Pages“-Dienst anderen Agenten anzubieten. Neben dem AMS steht auf jeder Agentenplattform mindestens ein *Directory-Facilitator* (DF) zur Verfügung, bei dem sich Agenten mit ihren Diensten registrieren können. Somit stellt ein DF einen „Yellow-Pages“-Dienst zur Verfügung, bei dem Agenten andere Agenten anhand von Diensten identifizieren können. Schließlich stellt der *Message-Transport-Service* (MTS) die Default-Kommunikationskomponente dar, die von den Agenten (auch verschiedener Plattformen) zum Austausch von Informationen genutzt wird [1]. Im Folgenden soll das von der FIPA spezifizierte Referenz-Modell als Grundlage für weitere Überlegungen fungieren.

## 3 Konsistenzsicherung durch Wartung

Im Rahmen der Erfüllung ihrer Aufgaben (definierten Ziele) können Agenten mit anderen Agenten kooperieren. Hierzu können Agenten u. a. lokal vorgehaltene Informationen verwen-

den anhand derer sie ihre Aktionen bestimmen. Diese Informationen können sich im Laufe der Zeit durch z. B. Aktualisierungen ändern und mit denen anderer Agenten differieren. Im schlimmsten Fall widersprechen sich die Informationen von Agenten, so dass ein inkonsistenter Zustand für das Gesamtsystem die Folge ist.

Die Aktionen, die zur Erhaltung und Wiederherstellung eines konsistenten Gesamtzustandes dienen, werden in diesem Beitrag unter dem Begriff der *Wartung* zusammengefasst. Zu ihnen zählen die (*De-*)*Registrierung* von Agenten bei der Wartungskomponente (s.u.), das Wahrnehmen relevanter Veränderungen sowie die *Notifikation* und die Aktualisierung registrierter Agenten. Neben diesen Aktionen können verschiedene *Dimensionen* der Wartung identifiziert werden (Wartungspriorität, Datenabhängigkeit, Skalierbarkeit, Verfügbarkeit, Wartungsanomalien) von denen die der *Wartungspriorität* und die der *Datenabhängigkeit* im Folgenden kurz erläutert werden.

### 3.1 Wartungspriorität

Jede Wartung, gleichgültig ob von Agenten oder anderen Einheiten, benötigt grundsätzlich Ressourcen und verursacht somit Kosten. Zur Kostenreduktion wird in diesem Beitrag der Ansatz vorgeschlagen, nicht jede, sondern nur als kritisch eingestufte Inkonsistenzen unmittelbar zu beseitigen. Eine derartige Einstufung erfordert die Definition einer *Wartungspriorität*, die festlegt, welche Dringlichkeit zur Wartung besteht. Je kritischer eine Situation ist, desto eine höhere Dringlichkeit zur Wartung liegt vor. Im Folgenden wird als *propagierender Agent* der Agent bezeichnet, der die neue (und korrekte bzw. aktuellste) Information besitzt, die er an die anderen, veraltete Information besitzenden und daher zu *wartenden Agenten* weiterleitet. Tatsächlich können bereits durch die nachfolgenden Beispiele verschiedene Stufen der Wartungsdringlichkeit (WD) identifiziert werden:

- hohe WD: Eine Situation in der der propagierende Agent die Information *Patient X hat eine Allergie gegen Eiweiß* besitzt, die zu wartenden Agenten dagegen die Information *Patient X hat keine Allergie gegen Eiweiß* besitzen, ist kritisch, da Agenten in einem solchen Fall z. B. zu einer Fehlmedikation führende Präparate empfehlen könnten.
- mittlere WD: In einer Situation, in der die Information eines propagierenden Agenten *Patient X hat keine Angst vor Spritze* ist und die zu wartenden Agenten die Information *Patient X hat Angst vor Spritze* besitzen, ist weniger kritisch. Im schlimmsten Fall würden die Anwender, die über noch nicht gewartete Agenten ihre Informationen beziehen, denken, dass der Patient Angst vor Spritzen hat und dementsprechend behutsamer mit ihm umgehen, als es sonst der Fall wäre.
- gerine WD: Eine (aus medizinischer Sicht) unkritische Situation liegt vor, wenn der propagierende Agent z. B. die Information *Vorname von Patient X ist Marie* besitzt und alle anderen Agenten die Information *Vorname von Patient X ist Maria*. Voraussetzung für eine unkritische Situation ist, dass der Vorname nicht zur Identifizierung von z. B. der Krankenakte herangezogen wird.

Ob und wie die Wartungspriorität automatisiert festgelegt und erkannt werden kann, ist derzeit noch nicht geklärt.

### 3.2 Datenabhängigkeit

Ein anderer interessanter Aspekt der Wartung ist die Dimension der Datenabhängigkeit. In den klassischen Fällen zur Replikation und zum View Maintenance [2, 3, 4, 6] werden keine Fälle von Datenabhängigkeiten berücksichtigt. Hierzu ein Beispiel: Agent B erhält von Agent A die Information *Patient X ist am 01.01.1993 gegen Tetanus geimpft worden*. Damit Agent B selbständig den Impfstatus überprüfen kann, lässt er sich von dem Agenten C (z. B. einem Impfpflichtagenten) berechnen, wann die nächste Impfung anstehen würde – in diesem Fall also am 01.01.2003. Nun erhält Agent A die Information, dass die Impfung nicht am 01.01.93 sondern am 01.01.99 statt gefunden hatte und das ursprüngliche Datum das Resultat einer Fehlerfassung war. Die Änderung muss dem Agenten B mitgeteilt werden. Wartungs-Mechanismen der klassischen Replikation würden nun den Agenten B entsprechend aktualisieren. Da die Abhängigkeit zwischen dem Agenten B und dem Agenten C nicht

explizit berücksichtigt wird, wäre nach der Aktualisierung des Agenten B die Wartungsaktion abgeschlossen. Die Folge wäre, dass obwohl der Patient X am 01.01.99 die letzte Impfung gegen Tetanus erhalten hatte, bereits am 01.01.2003 wieder eine Impfung von B zur Erhaltung des Impfschutzes empfohlen werden würde.

Zur Vermeidung einer solchen Situation muss der Agent B seine vorgehaltenen Informationen nach der im Rahmen der Wartungsaktion erfolgten Aktualisierung anpassen; dies gilt für *jede* Änderung. Ein naives Vorgehen würde hier dazu führen, dass der Agent u. U. sehr viele Informationen aktualisieren muss, die nicht mit den veränderten Informationen in Beziehung stehen. Um dieses zu vermeiden muss entweder der Agent B selbst Informationen über die Abhängigkeiten seiner Daten speichern oder es muss eine Wartungsinstanz existieren, die diese Informationen speichert. Weitere Untersuchungen werden zeigen, welche der Alternativen sinnvoller ist.

## 4 Die Wartungskomponente

Zur adäquaten Berücksichtigung der Wartungsproblematik und zur Sicherung der Konsistenz in agentenbasierten Informationssystemen wird in diesem Beitrag der Einsatz einer *Wartungskomponente* (MS) vorgeschlagen, die das derzeitige von der FIPA standardisierte Referenz-Modell erweitert (vgl. Abb. 1). Damit Agenten an einer Wartung teilnehmen können, müssen sie sich bei dieser Wartungskomponente registrieren. Hierzu senden sie die Nachricht `reg(agentenname, agententyp)` an die Wartungskomponente. Diese speichert in einer Liste, welcher registrierte Agent von welchem Wartungsagenten gewartet werden soll. Es existieren Wartungsagenten für jeden Agententypus. Registriert sich ein Agent eines neuen Agententypus bei der Wartungskomponente, so wird der entsprechende Wartungsagent des neu registrierten Agenten in einen Pool von Wartungsagenten aufgenommen. Der Pool der Wartungsagenten kann dabei unterschieden werden nach dem Pool von aktivierten und suspendierten Agenten. Agenten die zur Zeit nicht benötigt werden, können in den Pool der suspendierten Agenten überführt werden.

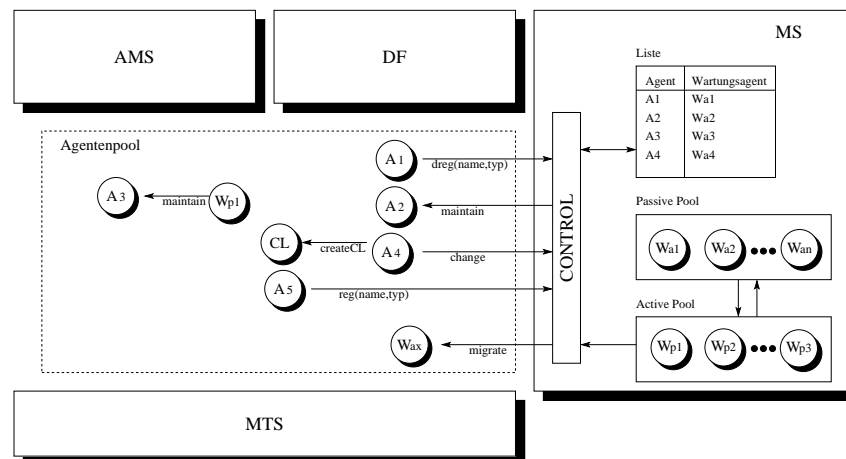


Abbildung 1: Wartungskomponente (MS)

Gelangen die Agenten in einen Zustand, der nicht mehr wartungsbedürftig ist bzw. entscheiden sie sich dazu, dass ihr Zustand nicht mehr aktualisiert werden muss, können sie sich bei der Wartungskomponente deregistrieren und werden aus der Menge der zu wartenden Agenten entfernt. Hierzu sendet der Agent die Nachricht `dereg(agentenname, agententyp)` an die Wartungskomponente. Diese entfernt den Agenten aus der Liste der zu wartenden Agenten. Sollte der Agent der letzte noch unter Wartung stehender Agent eines bestimmten Typus sein, so wird der zuständige Wartungsagent deaktiviert, zerstört oder suspendiert. In welchen Zustand der Agent zu überführen ist, ist zu untersuchen. Agenten, die sich bei der Wartungskomponente registriert haben, können mittels einer Suchfunktion gefunden werden, um neue Informationen von ihnen zu erhalten oder sie im Rahmen eines Updates zu

aktualisieren. Hierzu finden entsprechende Wartungsaktionen statt. Ob die Wartungskomponente selbst oder mehrere „ausströmende“ Wartungsagenten die Wartung übernehmen, ist ebenfalls noch nicht geklärt. Wenn sich die gespeicherten Werte eines Agenten ändern, informiert der Agent die Wartungskomponente, bei der er sich registriert hat. Die Informierung über die Änderung erfolgt entweder durch eine einfache Übermittlung der Veränderung bzw. des neuen Wertes oder durch die Generierung eines *ChangeInformation-Agenten*, der zur jeweiligen Wartungskomponente migriert. Nachdem eine Wartungskomponente über eine Veränderung informiert wurde, informiert sie die entsprechenden Wartungsagenten. Die Wartungsagenten übernehmen dann die Aufgabe der Wartung, d. h. sie stellen den Grad des Aktualitätsverlustes fest und die Dringlichkeit zur Wartung. Ist diese hoch werden sie unmittelbar zu den Agenten migrieren, die zu warten sind. Andernfalls erfolgt die Wartung zu einem definierten Zeitpunkt. Insgesamt muss eine Wartungskomponente entworfen werden, die neben der Bereitstellung der Managementfunktionen, eine Wartungsstrategie implementiert, die die genannten Wartungsdimensionen berücksichtigt.

## 5 Zusammenfassung und Ausblick

Zur Realisierung von Informationssystemen mit proaktiven und autonomen Eigenschaften wurde in diesem Beitrag der Einsatz von auf einer Agenten-Metapher basierenden Informationssystemen vorgeschlagen. Für solche agentenbasierten Systeme können aufgrund ihrer Verteiltheit von Architektur und Information Inkonsistenzen auftreten. Zur Sicherung und Wiederherstellung der Konsistenz des agentenbasierten Systems ist eine Wartung der von den Agenten lokal vorgehaltenen Informationen erforderlich. Es können dabei verschiedene Dimensionen der Wartung unterschieden werden, von denen die der Datenabhängigkeit sowie die der Wartungspriorität vorgestellt wurden. Zur Durchführung der Wartung wurde eine neue Komponenten vorgeschlagen, die das von der FIPA spezifizierte Referenz-Modell erweitert.

Folgende Arbeiten stehen aus: Da für den Agentenbereich bislang keine Arbeiten existieren, die sich explizit mit dem Problem der Wartung auseinandersetzen, sind Techniken und Strategien aus den Bereichen der klassischen Replikation und des View Maintenance auf Adaptierbarkeit zu untersuchen. Eine Diskussion und Bewertung aller Wartungsdimensionen ist durchzuführen. Insbesondere ist eine Definition von Konsistenzgraden und eines Prioritätenmodells erforderlich. Es ist zu klären, ob und wie die Wartungspriorität automatisiert erkannt werden kann. Für Agenten ist eine Klassifikation zu erarbeiten, die festlegt, welche Agenten z. B. neue Informationen propagieren dürfen. In diesem Zusammenhang müssen Zugriffsbeschränkungen berücksichtigt werden, d. h. existieren Agenten die nicht (z. B. aus Datenschutzgründen) über bestimmte Veränderungen informiert werden dürfen.

## Literatur

- [1] ‘Fipa agent management specification fipasc000011 and fipasc00023j’, Technical report, Foundation for Intelligent Physical Agents (FIPA), (December 2002).
- [2] Ashish Gupta, Dinesh Katiyar, and Inderpal Singh Mumick, ‘Counting solutions to the view maintenance problem’, in *Workshop on Deductive Databases, JICSLP*, pp. 185–194, (1992).
- [3] S. Weissman Lauzac and P. K. Chrysanthis, ‘Utilizing versions of views within a mobile environment’, in *Proceedings of the 9th International Conference on Computing and Information*, (June 1998).
- [4] Inderpal Singh Mumick, Dallen Quass, and Barinderpal Singh Mumick, ‘Maintenance of data cubes and summary tables in a warehouse’, in *SIGMOD Conference*, pp. 100–111, (1997).
- [5] Michael Wooldridge, *An Introduction to Multiagent Systems*, John Wiley & Sons (Chichester, England), 2002.
- [6] Yue Zhuge, Héctor García-Molina, Joachim Hammer, and Jennifer Widom, ‘View maintenance in a warehousing environment’, in *SIGMOD Conference*, pp. 316–327, (1995).