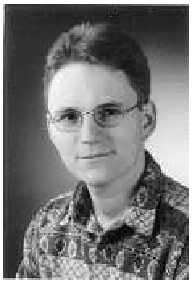
PIK — Praxis der Informationsverarbeitung und Kommunikation, 3/01, Saur Verlag, September 2001

Implementierung eines MASIF konformen Agentensystems — Die Mobile Agent System Architecture (MASA) —

Bernhard Kempter[1], Helmut Reiser[1], Harald Rölle[1], Gerald Vogt[2]

Munich Network Management Team
[1] Ludwig-Maximilians Universität München, [2] Technische Universität München
{kempter|reiser|roelle}@informatik.uni-muenchen.de vogt@in.tum.de

Dipl.–Inform. Bernhard Kempter, Studium der Informatik an der Technischen Universität München, Diplom 1999. Seit 1999 am Lehrstuhl Kommunikationssysteme und Systemprogrammierung (Prof. H.-G. Hegering) der Ludwig–Maximilians–Universität München beschäftigt und Mitglied des Münchner Netzmanagement Teams. Forschungsgebiete: Integriertes Netz– und Systemmanagement, Dienstmanagement insbesondere Lösung von QoS-Konflikten.



Dipl.–Inform. Helmut Reiser, Studium der Informatik an der Technischen Universität München, Diplom 1997. Seit 1997 am Lehrstuhl Kommunikationssysteme und Systemprogrammierung (Prof. H.-G. Hegering) der Ludwig–Maximilians–Universität München beschäftigt und Mitglied des Münchner Netzmanagement Teams. Forschungsgebiete: Integriertes Netz– und Systemmanagement, Sicherheitsaspekte im IT–Management und in Mobilen Agentensystemen. Mitglied bei IEEE und ACM.

Dipl.–Inform. Harald Rölle, Studium der Informatik an der Technischen Universität München, Diplom 2000. Seit 2000 am Lehrstuhl Kommunikationssysteme und Systemprogrammierung (Prof. H.-G. Hegering) der Ludwig–Maximilians–Universität München beschäftigt und Mitglied des Münchner Netzmanagement Teams. Forschungsgebiete: Integriertes Netz– und Systemmanagement, speziell Dienstmodellierung und Quality of Service.

Dipl.–Inform. Gerald Vogt, Studium der Informatik an der Universität Stuttgart. Diplomarbeit am Distributed Systems Technology Center (DSTC) in Brisbane, Australien. Diplom 1997. 1997/98 Consultant bei Borland, USA. Seit 1998 Stipendiat am Graduiertenkolleg „Kooperation und Ressourcenmanagement in verteilten Systemen" an der TU München am Lehrstuhl von Prof. H.-G. Hegering. Forschungsgebiete: Sicherheitsmanagement, IT-Management mit Mobilen Agenten insbesondere unter Sicherheitsaspekten.

Zusammenfassung

In einer global agierenden, arbeitsteiligen Wirtschaft sind große vernetzte IT–Systeme unabdingbare Infrastrukturen. Ihr sicheres und schnelles Funktionieren ist absolut unternehmenskritisch. Dem Management solcher großen, heterogenen und dynamischen IT–Infrastrukturen kommt daher eine wachsende Bedeutung zu.

Traditionelle, zentralisierte Managementsysteme sind oft nicht in der Lage mit einer hohen Dynamik adäquat umzugehen. Sie lassen sich auch nicht auf beliebig grosse Infrastrukturen skalieren. Auf Mobilen Agenten basierende Managementsysteme erscheinen als vielversprechende Alternative. Ein solches Managementsystem ist eine verteilte, dezentrale Anwendung, die der hohen Dynamik und dem Bedarf nach Flexibilität und Anpassbarkeit im laufenden Betrieb sowie lokaler Autonomie gerecht werden kann.

Aus diesem Grund wurde die Mobile Agent System Architecture (MASA) als Forschungsprototyp entwickelt. MASA basiert auf der MASIF–Spezifikation. In diesem Artikel wird die Agentenplattform MASA vorgestellt und die MASIF–Spezifikation vor dem Hintergrund praktischer Implementierungs– und Anwendungserfahrungen bewertet und dabei die Grenzen der von MASIF versprochenen Interoperabilität aufgezeigt.

Schlüsselwörter:
MASIF, Mobile Agenten, kooperierende Software Agenten, IT–Management

1 Einleitung

Mit der wachsenden Vernetzung durch die weltweite Ausbreitung des Internets und mit der immer größeren Abhängigkeit der Unternehmen von ihren Netzen, wachsen auch die Anforderungen an das IT–Management. Während früher Ausfälle von Teilen der IT–Systeme für Unternehmen noch begrenzte, absehbare Folgen hatten, ist inzwischen die Abhängigkeit von der Leistungsfähigkeit und Zuverlässigkeit dieser Systeme stark gestiegen. Neue Herausforderungen entstehen auch durch die große Zahl von Fusionen der letzten Jahre: Zwei vorher getrennte, große, völlig unterschiedliche IT–Systeme müssen geeignet in eine neue, gemeinsame IT–Infrastruktur überführt werden, um Synergieeffekte voll ausnutzen zu können. Auf diese Weise entstehen global agierende Konzerne mit entsprechend umfassenden und komplexen IT–Infrastrukturen. Sie unterliegen einem stetigen Wandel in Bezug auf Komponenten, Topologien und Technologien.

Bedingt durch die essentielle Bedeutung der IT–Systeme als unternehmenskritischer Faktor, ihrer Dynamik und nicht zuletzt durch ihre immer kürzeren Innovationszyklen ergeben sich neue Anforderungen an das IT–Management. Viele Unternehmen, wie zum Beispiel Finanz– und Versicherungsdienstleister oder Just–in–time–Produzenten, haben allerhöchste Anforderungen an die Verfügbarkeit und Sicherheit ihrer IT–Infrastruktur. Ausfälle, auch kurzzeitige, haben enorme finanzielle Folgen. Es wird beispielsweise geschätzt, dass für eine Bank ein Ausfall der Netzinfrastruktur bzw. der Kernanwendungen nach nur zwei Tagen bereits den Ruin bedeuten kann.

Die Aufgaben des IT–Managements erstrecken sich von der Planung, über die Bereitstellung, die Konfiguration, den Betrieb, die Wartung, die Überwachung, die interne und externe Abrechnung bis hin zur problemlosen und zeitgerechten Migration zu neuen Technologien [HAN 99a].

Die im Moment auf dem Markt befindlichen, traditionellen Managementsysteme sind nicht in der Lage, die neuen Anforderungen zu erfüllen. Sie basieren meist auf Managementarchitekturen, in denen eine feste Managementfunktionalität einer zu managenden Ressource durch sogenannte Agenten repräsentiert wird. Diese Management–Agenten, die im Allgemeinen sehr einfach gehalten sind, dürfen nicht mit anderen Agentenbegriffen (z.B. Intelligente Agenten, ...) verwechselt werden, wie sie etwa aus der KI bekannt sind. Der Begriff Management-Agent steht für einen Teil der Ressource, der dem Managementsystem den Zugriff über eine genormte Schnittstelle ermöglicht. Die etablierten Systeme (die z.B. auf SNMPv1 [RFC 1157] basieren) erlauben darüber hinaus vielfach keine dynamischen Anpassungen zur Laufzeit des Systems und erfüllen auch die von Seiten der Sicherheit gestellten Ansprüche nicht.

In Forschungskooperationen mit Unternehmen wie BMW, Siemens oder DeTeSystem hat sich gezeigt, dass die Wirtschaft sehr großen Bedarf an flexiblen, dynamisch erweiterbaren und organisationsgrenzen–überschreitenden Managementlösungen hat [HaRe 00, Hojn 99, Knoe 99]. Heute werden Managementsysteme benötigt, die die Möglichkeit bieten, Funktionalität auch zur Laufzeit zu ändern und zu erweitern. Es soll auch möglich sein, Managementfunktionen vom Managementsystem eines Providers einfach auf den Kundenrechner zu übertragen und dort auszuführen. Von den Agenten, die Managementfunktionen implementieren, wird ein hoher Grad an Autonomie erwartet, damit sie selbst bei Verlust der Verbindung zum Managementsystem selbständig weiterarbeiten können. Ebenso sollen sie die Möglichkeit zur gemeinsamen und kooperativen Problemlösung besitzen und die Heterogenität der Systeme so weit wie möglich verschatten. Bei geänderten Anforderungen muss es eine einfache Möglichkeit zur Aktualisierung der Agenten geben. Für Managementanwendungen ist auch die Möglichkeit der Einbindung von bestehender (Legacy–) Software und die Interoperabiltät von entscheidender Bedeutung. Die Einhaltung von Standards ist eine Grundvoraussetzung dafür. Die Sicherheit des gesamten Systems ist — insbesondere weil Domänengrenzen überschritten werden — von entscheidender Bedeutung für die Akzeptanz und die Einsetzbarkeit in der Praxis.

Diese Erkenntnisse bekräftigt bereits seit längerem auch die Forschung. Vorschläge wie Management–by–Delegation [Gold 96, Schö 97], Flexible Agents [Moun 97], Script–MIB [LeSc 99], erweiterbare Agenten [Reis 99] usw. versuchen, Managementarchitekturen um flexiblere, dynamische und verteilte Elemente so zu erweitern, dass Programmcode bzw. ganze Agenten im Netz zu verschiedenen Knoten transportiert werden können, um dort Aufgaben zu erfüllen.

Ergänzt man die Fähigkeit autonom zu handeln, zu kommunizieren und zu kooperieren, um die Möglichkeit sich frei im Netzwerk zu bewegen, so entstehen autonome, mobile Software-Agenten, sogenannte Mobile Agenten [BPW 98]. Mit ihnen wird es möglich, Managementarchitekturen zu erstellen, die dem IT–

Management die geforderte Flexibilität geben und ihm eine sehr viel größere Dynamik verleihen. Es können notwendige Konfigurationsänderungen der Infrastrukturen einfacher und effizienter durchgeführt werden, wenn zum Beispiel neue Systeme oder Anwendungen eingeführt werden. Darüber hinaus können während des gesamten Betriebs Agenten Überwachungsaufgaben übernehmen und eventuell später notwendige Anpassungen vollautomatisch vornehmen.

Um Vorteile und Potentiale der Mobilen Agententechnologie für das Einsatzgebiet IT–Management zu untersuchen und zu bewerten, wurde die Agentenplattform MASA entwickelt.

Im folgenden Abschnitt werden die grundlegenden Konzepte und Unterschiede der beiden Mobilen Agenten Spezifikationen MASIF und FIPA vorgestellt und verglichen. Kapitel 3 beschreibt die Mobile Agent System Architecture (MASA), die MASIF als Basis verwendet. Im Kapitel 4 werden die Stärken und Schwächen der MASIF Spezifikation aus Sicht der Entwicklung und des Einsatzes von MASA diskutiert. Kapitel 5 fasst die Arbeit zusammen und gibt einen Ausblick auf weitere Forschungsfragestellungen.

2 MASIF und FIPA im Vergleich

Bestehende Plattformen für Mobile Agenten unterscheiden sich in den Konzepten, in ihrer Architektur und in der Implementierung erheblich voneinander. Diese Unterschiede erschweren oder verhindern eine Interoperabilität, also die Möglichkeit, dass Mobile Agenten zwischen Agentensystemen verschiedener Hersteller migrieren können. Diese Interoperabilität ist aber für eine rasche Verbreitung und Etablierung von Mobilen Agententechnologien essentiell.

Mit Mobilen Agententechnologien beschäftigen sich ernsthaft nur zwei Spezifikationen. Diese beiden, die Mobile Agent System Interoperability Facility (MASIF) und die Spezifikation der Foundation for Intelligent Physical Agents (FIPA), werden im Folgenden vorgestellt, um sie anschließend zu vergleichen zu bewerten.

2.1 MASIF

Die Object Management Group (OMG)[1], ein Konsortium von über 800 Mitgliedern mit dem Ziel, Standards in der Computer–Industrie zu etablieren, hat 1997 die Spezifikation *Mobile Agent System Interoperability Facility (MASIF)* veröffentlicht, welche 1998 überarbeitet wurde [MASIF].

Um Interoperabilität zu erreichen war es aus Sicht der OMG notwendig, folgende Bereiche zu standardisieren:

- die Terminologie, um ein einheitliches Verständnis von Begriffen zu erreichen,
- die Namensgebung für Agenten und Agentensysteme, damit eine eindeutige Identifizierung ermöglicht wird,
- die Lokalisierung von Agenten und Agentensystemen, um eine Kommunikation zu etablieren,
- den Agententransfer, zur Unterstützung der Mobilität von Agenten und
- das Agentenmanagement, um den Lifecycle eines Agenten zu verwalten.

Im Weiteren werden diese Bereiche detailliert beschrieben.

Ein *Agent*, im Sinne von MASIF, ist ein ablauffähiges Programm, welches autonom im Auftrag einer *Authority* (Person oder Organistion) handelt. Dabei gibt es zwei Ausprägungen von Agenten. Ein *Stationary Agent* ist ein Agent, der nur auf dem Agentensystem ausgeführt wird, auf dem er gestartet wurde. Im Gegensatz dazu kann ein *Mobile Agent* von Agentensystem zu Agentensystem migrieren, um seine gestellte Aufgabe zu erfüllen.

Ein *Agent System* stellt die Laufzeitumgebung für Agenten dar, besitzt einen eindeutigen *Agent System Type* und wird ebenfalls von einer *Authority* gestartet. Agenten befinden sich innerhalb des Agentensystems in einer Ausführungsumgebung, den sogenannten *Places*. Ein Place kann eine Umgebung sein, in der z. B. bestimmte Zugriffsrechte auf Ressourcen gelten. Abbildung 1 zeigt die Einbettungsbeziehungen der eingeführten Begriffe.

Mehrere Agentensysteme derselben Authority können zu einer *Region* zusammengefasst werden. Agentensysteme kommunizieren über eine Kommunikationsinfrastruktur miteinander. Diese Infrastruktur wird durch die Common Object Request Broker Architecture (CORBA) [CORBA 2.2] bereitgestellt.

Agenten und Agentensysteme besitzen jeweils einen eindeutigen Identifikator bzw. Namen, der immer aus folgenden Teilen besteht:

- *Authority*: Name der verantwortlichen Person oder Organistion
- *Identity*: ein eindeutiger Wert innerhalb des Verantwortungsbereichs einer Authority

[1] `http://www.omg.org`

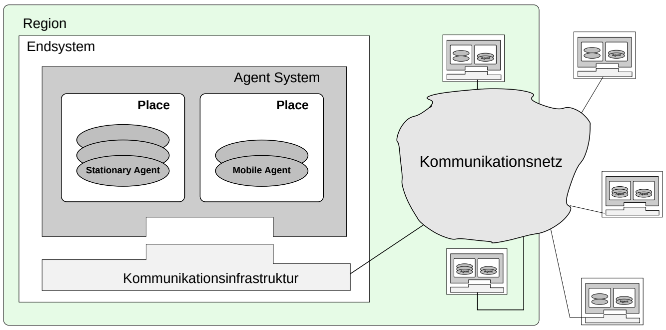


Abbildung 1: MASIF–Basismodell

- *Agent System Type*: von der OMG vergebene, weltweit eindeutige Nummer

Der so zusammengesetzte Name muss global eindeutig sein.

Auch jeder Place erhält einen Namen. Dieser muss nur innerhalb des Agentensystems eindeutig sein. Zur Zuweisung von Namen für Places werden keine genaueren Angaben gemacht.

Ein Agent wird nicht direkt adressiert, sondern immer über die Adresse des Agentensystems und den Place lokalisiert, in dem er ausgeführt wird.

Die Adresse des Agentensystems ist entweder eine

- URI[2], welche einen CORBA Namen enthält, oder eine
- URL[3], die eine Internet–Adresse darstellt.

Die Lokalisierung, also die Abbildung von Namen auf Adressen übernimmt der sogenannte *MAFFinder*. Er bietet über eine CORBA–Schnittstelle Operationen an, die diese Abbildung vornehmen.

Es wird ein Lifecycle für Agenten spezifiziert, der u. a. die Möglichkeit der Serialisierung und damit die Fähigkeit zur Migration der Daten sowie des Code des Agenten ermöglicht. Die konkrete Realisierung dieser Verfahren ist nicht Teil der Standardisierung, sondern bleibt der Implementierung überlassen.

Jedes Agentensystem stellt an seiner CORBA–Schnittstelle Operationen zur Verfügung, die es ermöglichen, den gesamten Lifecycle eines Agenten, einschließlich seiner Migration, zu verwalten.

2.2 FIPA

Die Foundation for Intelligent Physical Agents (FIPA) [FIPA 98] ist eine non-profit Vereinigung von Firmen und Forschungseinrichtungen (derzeit 47 Mitglieder). Ihr Ziel ist die Spezifikation einer generischen Agententechnologie, um einen hohen Grad an Interoperabilität zwischen agentenbasierten Anwendungen zu erreichen. Die Konzepte dieser Spezifikation werden im Folgenden vorgestellt.

Ein Agent ist ein Akteur, der auf einer Agentenplattform ausgeführt wird und intelligent im Sinne der KI ist. Auf der Agentenplattform ist das *Agent Management System* integriert, das für das Management des Agenten–Lifecycle verantwortlich ist. Weiterhin besteht die Agentenplattform aus einem *Directory Facilitator*, einem Verzeichnisdienst, bei dem ein Agent seine Dienste registrieren lassen bzw. nach Diensten anderer Agenten suchen kann. Über den *Agent Communication Channel* kommunizieren Agenten, welche sich auf unterschiedlichen Agentenplattformen befinden. Zum Nachrichtenaustausch wird innerhalb einer Agentenplattform ein *Internal Platform Message Transport* vorgeschlagen, aber nicht weiter spezifiziert.

In einer *Agent Domain* werden Agenten zu Gruppen zusammengefasst. Eine Gruppe wird durch ein Verzeichnis innerhalb eines Directory Facilitators dargestellt. Jeder Agent muss mindestens einer Agent Domain angehören.

Zur Inter–Agentenkommunikation definiert FIPA die Agent Communication Language (ACL), einer an KQML (Knowledge Querying and Manipulation Language [KQML 97]) angelehnten Sprache, mit deren Hilfe Agenten Informationen austauschen können. Die Sprache besitzt eine große Mächtigkeit und wurde entwickelt, damit KI Komponenten Wissen teilen und austauschen können. ACL soll es Agenten ermöglichen „semantisch nützliche“ Informationen („semantically usefull information“ [FIPA 00]) ableiten können, ohne sich a–priori über das genaue Verständnis einer Sprache einig zu sein, die zur Kommunikation verwendet wird. Dazu wird innerhalb einer ACL Nachricht auf eine oder mehrere Ontologien verwiesen, mit deren Hilfe es möglich wird, den semantischen Gehalt der Nachricht abzuleiten.

2.3 Vergleich von MASIF und FIPA

Die beiden Spezifikationen werden nun hinsichtlich ihrer Eignung zu Managementzwecken und der Unterstützung der Implementierung betrachtet.

FIPA setzt mit seiner Spezifikation den Fokus auf intelligente Agenten im Sinne der KI. Die Kommunikation zwischen den Agenten wird durch die, doch sehr komplexe Agent Communication Language unterstützt. Für Managementanwendungen ist diese Komplexität

[2] Uniform Resource Identifiers
[3] Uniform Resource Locators

und Mächtigkeit der Kommunikation i.d.R. nicht erforderlich und zum Teil sogar kontraproduktiv. Im Management geht es nicht primär darum „Wissen“ zu teilen, sondern Informationen über Zustände in den zu verwaltenden Komponenten abzufragen oder zu verändern. Es ist auch nicht erforderlich, sich mit Hilfe von ACL über die Semantik einer Sprache zu verständigen, mit der die Managementinformation übertragen wird. Die Semantik der Managementinformation wird im Informationsmodell der Managementarchitektur festgelegt. Für ACL, als Teilmenge der Sprache KQML, wird kaum Werkzeugunterstützung geboten. So muss z.B. für Agenten, die mittels ACL oder KQML kommunizieren sollen, ein eigener Parser geschrieben werden, um die Sprache zu interpretieren. Über die Kommunikationsinfrastruktur, die zur Übertragung der ACL Nachrichten verwendet werden soll, werden keine Aussagen gemacht.

MASIF bietet im Gegensatz dazu, durch die Abstützung auf CORBA, eine ausgereifte Kommunikationsinfrastruktur, die es ermöglicht, dass Agenten auf verschiedenen Rechensystemen ortstransparent interagieren können. Dazu werden Aufrufschnittstellen der Agenten, wie auch anderer CORBA Objekte, in der einheitlichen, programmiersprachen– und systemarchitekturunabhängigen Interface Definition Language (IDL) beschrieben. Für CORBA gibt es eine breite Werkzeugunterstützung. So können z.B. Compiler aus den IDL Schnittstellenbeschreibungen Code–Rahmen für die verschiedensten Implementierungssprachen erzeugen, die bereits alle Komponenten für die Kommunikation enthalten. Die Abstützung auf CORBA als Middleware vereinfacht auch die Integration von bestehenden Managementanwendungen (Legacy–Anwendungen) über Betreiber– und Betriebssystemgrenzen hinweg. Altanwendungen werden dazu mittels IDL und entsprechenden Proxy–Objekte gekapselt. In [Kell 98] wurde gezeigt, dass sich CORBA sehr gut eignet, um unter einem sogenannten Umbrella–Management alle bestehenden und neuen Management–Anwendungen innerhalb eines Unternehmens zusammenzufassen, und damit die, zum Teil erheblichen, Investitionen in vorhandene Systeme zu sichern. CORBA ist eine standardisierte und integrierende Kommunikationsplattform, die neben der erheblichen Anzahl der OMG–Mitglieder auch eine sehr viel breitere kommerzielle Unterstützung erfährt als dies bei FIPA der Fall ist.

Aus diesen Gründen wurde für die Implementierung der Mobile Agent System Architecture die MASIF–Spezifikation und nicht FIPA verwendet.

3 Die Mobile Agent System Architecture

In diesem Kapitel wird nun mit der Mobile Agent System Architecture (MASA) eine Implementierung eines zu der im vorangegangenen Kapitel vorgestellten MASIF–Spezifikation konformen Agentensystems vorgestellt.

Grundgedanke bei Design und Implementierung von MASA ([Kemp 98]) war es, ein hinsichtlich des Agentensystems völlig plattform– und betriebssystemunabhängiges System zu schaffen. Diese Anforderung ergab sich unmittelbar aus der im Einsatzgebiet des Netz- und Systemmanagements immanent vorhandenen Heterogenität der zu managenden Systeme und Komponenten. Die Agentensysteme sollen auf dieser heterogenen Infrastruktur eine homogene Ausführungsplattform für Mobile Agenten bilden. Das Einsatzgebiet IT-Management bedingt aber auch, dass es nicht möglich ist und sein wird, alle Teile eines Managementsystems und insbesondere die Managementagenten vollkommen plattformunabhängig zu gestalten.

Hinsichtlich des Basissystems bietet Java, neben einer guten Werkzeugunterstützung, die Möglichkeit viele Plattformen ohne spezifische Anpassungen zu erreichen und wurde vor allem deshalb als Implementierungssprache gewählt. Weiterhin ist durch die MASIF–Spezifikation mit CORBA bereits eine ebenso leistungsfähige wie plattformunabhängige Middleware gegeben. Die Benutzeroberflächen von Agentensystem und Agenten sollen über einen Web–Browser erreichbar sein. Durch die Oberflächenintegration mit anderen (Web-basierten) Managementsystemen wird damit ein erster Schritt in Richtung eines vollständig integrierten Managements [HAN 99a] realisiert.

Abbildung 2 stellt die Kommunikationsbeziehungen der Komponenten einer MASA–Region dar. Alle Komponenten von MASA kommunizieren über CORBA (konkret über das Internet Inter ORB Protocol (IIOP)). In MASA besitzt jeder Agent und auch das Agentensystem selbst ein Applet, das die Benutzerschnittstelle realisiert. Ein Benutzer, der ein Applet laden möchte, baut eine HTTP–Verbindung zum HTTP–Server–Agenten des Agentensystems auf, der das entsprechende Applet zur Verfügung stellt. Sobald das Applet im Browser gestartet wurde, kommuniziert dieses wieder über IIOP mit „seinem“ Agenten.

Neben der Konformität zu MASIF wird in MASA besonders auf Sicherheitsaspekte geachtet. Während das generelle Anwendungsszenario Management bereits hohe Sicherheitsanforderungen stellt, wird die Problematik durch die Mobilität der Agenten noch-

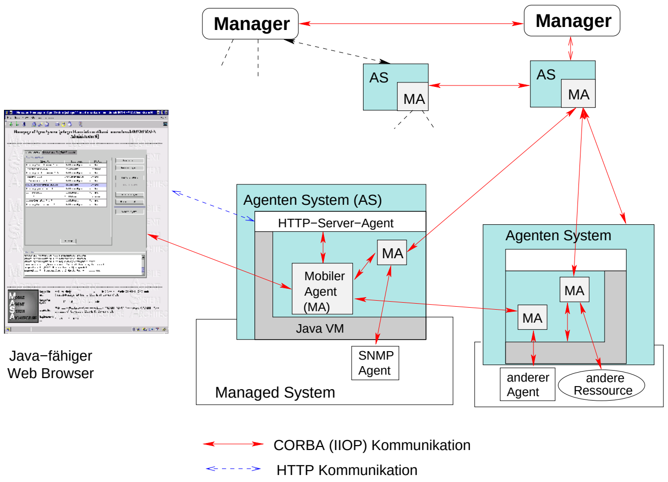


Abbildung 2: Kommunikation in MASA

mals verschärft, insbesondere wenn man Mobile Agenten domänenübergreifend im Management einsetzt [ReVo 00a]. Den speziellen Anforderungen wird in MASA durch eine eigene Sicherheitsarchitektur [Roel 99a] Rechnung getragen, die in ihrem Modell eine feingranulare Unterscheidung aller beteiligter Entitäten liefert und in dessen Implementierung Standards wie Transport Layer Security (TLS) bzw. Secure Socket Layer (SSL) und X.509 zum Einsatz kommen. Auf dieser Basis wird jegliche Kommunikation zwischen den Entitäten verschlüsselt und die Kommunikationspartner können sich gegenseitig authentisieren (CORBA–SSL und HTTPS).

Die Implementierungsarchitektur von MASA, dargestellt in Abb. 3, gliedert sich in vier Hauptschichten. Die unterste Schicht ist das Betriebs– und Transportsystem. Diese sollen als gegeben angesehen und somit nicht weiter betrachtet werden.

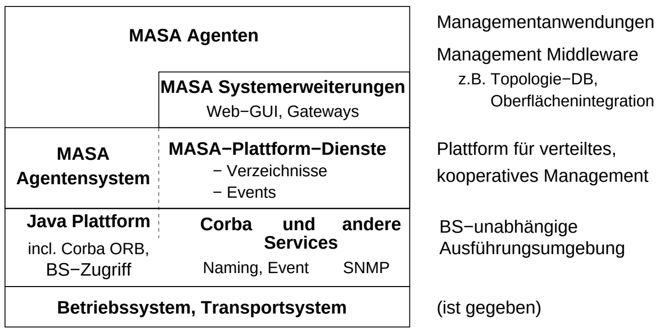


Abbildung 3: Schichtenarchitektur von MASA

Darüber liegt die Abstraktionsschicht für das Betriebs– und Transportsystem, welche die plattformunabhängige Basis für MASA bereitstellt. Darauf aufbauend liegt das MASA–Agentensystem, welches die eigentliche Plattform für das kooperative und verteilte Management darstellt. In der obersten Schicht befinden sich die MASA–Agenten sowie MASA–Systemerweiterungen, die schließlich die konkreten Managementanwendungen bzw. Management–spezifische Middleware–Komponenten realisieren. Im Folgenden werden diese einzelnen Schichten näher betrachtet.

3.1 Plattformunabhängige Ausführungsumgebung

Mittels Java wird die Abstraktion von speziellen Betriebssystemen und Plattformen realisiert. Dies wird erreicht, indem der Java Quellcode in einen plattformunabhängigen Zwischencode (Bytecode) übersetzt wird, und dieser durch die Java Virtual Machine (JVM), einem auf die konkrete Plattform angepassten Interpreter, ausgeführt wird (Abb. 2).

Zur Sprache Java gehört ein umfassender Satz von Entwicklungswerkzeugen und Bibliotheken, das Java Development Kit (JDK). Hierdurch wird die plattformunabhängige Entwicklung weiter unterstützt. In MASA wird dabei die Version 1.2 des JDK eingesetzt.

Die Middleware des Gesamtsystems wird durch einen, bereits vollständig in Java implementierten und damit plattformunabhängigen CORBA Object Request Broker (ORB) realisiert. Dem ORB zur Seite stehen dabei diverse CORBA Services, insbesondere der CORBA Naming Service und der CORBA Event Service. Mittels des Naming Service wird ein systemweiter Verzeichnisdienst realisiert. Der Event Service stellt einen globalen Mechanismus zur asynchronen, Pull– oder Push–basierten Ereigniskommunikation bereit.

3.2 MASA–Agentensystem

Das Agentensystem stellt die Ausführungsumgebung der Mobilen Agenten dar. Es wurde vollständig in Java implementiert, womit bereits das Agentensystem die gleiche Portabilität wie die Agenten besitzt.

Hauptaufgabe des Agentensystems ist die Verwaltung der Mobilen Agenten und die Durchsetzung von Sicherheitseigenschaften. Um die Flexibilität der Implementierung zu steigern und Erweiterungen zu erleichtern, ist es durch die im Folgenden beschriebenen Java–Klassen modularisiert:

- Klasse AgentSystem

  AgentSystem ist der zentrale Teil des Agentensystems und implementiert alle der in MASIF definierten Funktionen der Interfaces MAFAgentSystem und MAFFinder und stellt entsprechende

CORBA Schnittstelle bereit. Die darin enthaltene Funktionalität umfasst:

  - Erzeugen, Migrieren und Terminieren von Agenten
  - Anhalten und Fortsetzen der Ausführung von Agenten
  - Registrieren und Entfernen von Agenten und Agentensystemen aus dem MASIF–Namensverzeichnis
  - Abfragen und Durchsuchen des MASIF–Namensverzeichnisses

  Um neben MASA–Agenten weitere Agententypen unterstützen zu können, werden die konkreten Implementierungen in für die jeweiligen Agententypen spezifischen Manager–Klassen realisiert. AgentSystem übernimmt somit hauptsächlich die Aufgabe eines Dispatchers und stellt hierdurch die zentrale Stelle für Autorisierungsmaßnahmen dar. Weiterhin wird dem Benutzer eine Kommandozeile für Basisoperationen angeboten.

- Klasse MASAFinder

  Der im Rahmen von [Gigl 00] entwickelte MASA-Finder implementiert den in MASIF definierten MAFFinder, das MASIF–Namensverzeichnis. Sie stützt sich dabei auf den CORBA Naming Service.

- Klasse AgentManager

  AgentManager realisiert den Manager für MASA–Agenten und implementiert die von AgentSystem angebotenen MASIF–Funktionen für Agenten des Typs MASA. Neben den MASIF–Funktionen werden zusätzliche, MASA–spezifische Funktionen implementiert. Diese dienen zur Vereinfachung der Kommunikation zwischen Agenten, der Speicherung von serialisierten Agenten in Dateien, sowie der Lieferung umfangreicher Statusinformationen.

- Klasse AgentSecurityManager

  Vom AgentSecurityManager werden Authentisierungs– und Autorisierungsaufgaben übernommen, die in der MASA–Sicherheitsarchitektur festgelegt sind. Dazu gehören u.a. die Authentisierung von Mobilen Agenten und, basierend auf Policies, die Kontrolle ihrer Zugriffe auf das Endsystem, sowie auf eigene/fremde Agentensysteme und eigene/fremde Agenten.

- Klasse AgentSystemCertManager

  Der AgentSystemCertManager erteilt und verwaltet kryptographische Zertifikate im Rahmen der MASA–Sicherheitsarchitektur.

Der für eine Managementanwendung wichtigen und notwendigen Forderung, bestehende Systeme relativ leicht integrieren zu können, wird mit der Trennung in systemspezifische Manager Rechnung getragen. Als Beispiel für einen weiteren Manager ist das in [Bran 99] realisierte Voyager Gateway zu nennen. Damit ist es möglich, Mobile Agenten der Voyager 2.0 Plattform [Voyager 2.0] unter der Kontrolle von MASA auszuführen.

3.3 MASA–Systemerweiterungen

Die Benutzerschnittstelle für alle Agenten ist in Form von Java–Applets realisiert. Um diese in einen Browser laden zu können, wird ein HTTP–Server benötigt, der selbst als Agent implementiert wurde.

Die Steuerung des Agentensystems selbst erfolgt ebenfalls über ein Applet (Abb. 4). Es ermöglicht dem Benutzer neben der Nutzung der MASIF–Basisfunktionen folgende weitere Möglichkeiten:

- Abfragen von Statusinformationen über Agenten und das Agentensystem
- Anzeige der Zertifikate von Agenten und Agentensystem, u.a. ihrer Implementierer und Eigentümer
- Durchsuchen von Repositories startbarer Agenten
- Beenden des Agentensystems

Neben der Steuerung eines einzelnen Agentensystems gibt es durch den RegionManagementAgent [Gerb 99] auch eine GUI, welche eine gesamte Region betrachtet. Ein Beispiel für eine managementspezifische Systemerweiterung ist der in [Roel 99b] implementierte CORBA Topology Service, ein globaler Dienst zur Speicherung und Abfrage von Topologieinformationen.

3.4 MASA–Agenten

MASA unterscheidet MASIF–konform zwischen mobilen und stationären Agenten. Stationäre Agenten können nicht migriert werden und implementieren Methoden, die nur auf dem Start–System des Agenten ausgeführt werden dürfen. Der HTTP-Server Agent ist ein Beispiel hierfür. Die Mobilitätseigeschaften eines Agenten werden zur Implementierungszeit festgelegt.

Weiterhin differenziert MASA zwischen multiplen und exklusiven Agenten. Multiple Agenten dürfen in beliebiger Anzahl vorhanden sein, während von einem exklusiven Agenten höchstens eine Instanz innerhalb einer Region möglich ist. Die Attributierung multipel/exklusiv kann zur Laufzeit des Systems, aber nur vor dem Start des jeweiligen Agenten, festgelegt werden.

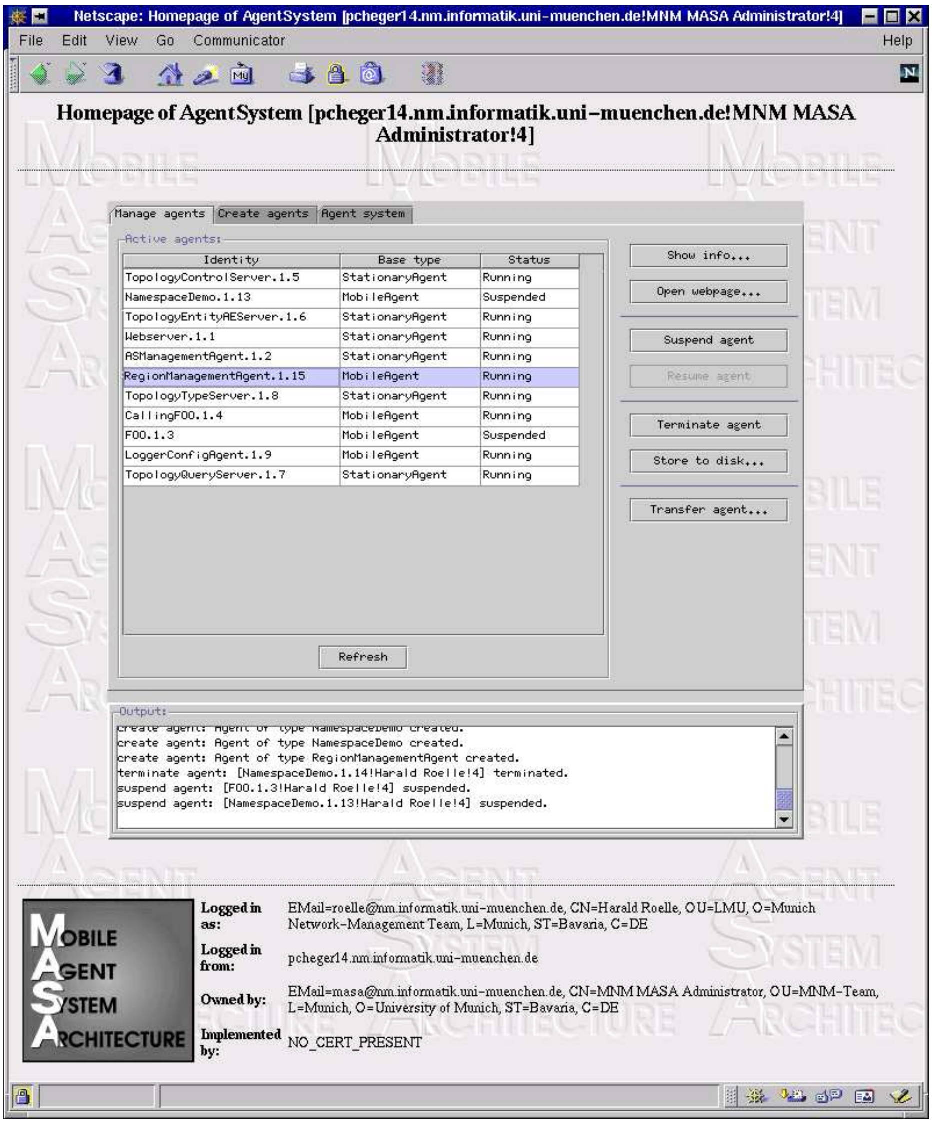


Abbildung 4: Applet des Agentensystems

Der Bytecode des Agenten und seine Benutzeroberfläche (Applet) werden gemeinsam in eine Java Archive (jar) Datei gepackt. Diese wird dann digital signiert, womit im Rahmen der Sicherheitsarchitektur der Implementierer eines Agenten festgelegt ist.

Im Lebenszyklus eines Agenten (Abb. 5) ist die Migration besonders zu beachten. Die in MASA realisierte Semantik besteht darin, dass Agenten, bevor sie migriert werden können, an einer definierten Stopp–Funktion angehalten werden, um anschließend auf dem Zielsystem in einer ebenfalls festen Funktion fortgesetzt zu werden. Dies bedeutet, dass Agenten zwar ihre Attribut–Belegung behalten, ihre Ausführung jedoch nicht an jedem beliebigen Punkt innerhalb ihres Instruktionspfades unterbrochen und dann linear fortgesetzt werden kann. Die Stopp–Funktion bietet dem Agenten die Möglichkeit Vorbereitungen für die Migration zu treffen, und muss durch ihn angesprungen werden, unabhängig davon, ob die Migration durch den Agenten selbst oder durch das Agentensystem initiiert wurde. Durch den fehlenden Zugriff auf Interna der JVM kann eine voll transparente Migration, d.h. nahtlose Fortsetzung des Programmflusses, nicht realisiert werden. In [Funf 98] werden auch einige Anwendungsfälle angegeben, bei denen eine voll transparente Migration überhaupt nicht realisierbar ist. Als Beispiele seien hier nur lokale Referenzen des Mobilen Agenten, wie z.B. geöffnete Dateien oder Netzwerkverbindungen genannt.

Des Weiteren können MASA Agenten persistent gespeichert werden, was mit einer zur Migration äquivalenten Semantik geschieht.

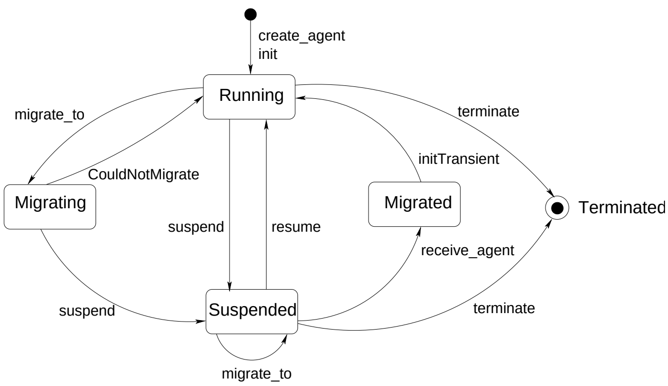


Abbildung 5: Lebenszyklus eines MASA–Agenten

Die Kommunikation zwischen Agenten sowie mit Agentensystemen geschieht, auch bei lokaler Kommunikation, ausschließlich über CORBA, genauer über CORBA–SSL. Damit können für alle Agenten, Agentensysteme und Benutzer Zertifikate ausgestellt werden. CORBA–SSL erlaubt es damit an allen Schnittstellen eine Authentisierung des Kommunikationspartners durchzuführen. Aufbauend auf die Authentisierung aller Komponenten des Systems lassen sich umfangreiche Autorisierungsmöglichkeiten realisieren.

Beispiel für verschiedenste, bereits implementierte, managementspezifischen Agenten finden sich in [Demm 99, Grus 99, Heil 99, Radi 98].

4 Erfahrungen bei der Umsetzung und Bewertung von MASIF

Mit MASA wurde eine Plattform für Mobile Agenten entwickelt, welche die MASIF–Spezifikation implementiert. Im Folgenden werden die Erfahrungen dargestellt, die bei der Umsetzung der Spezifikation in eine konkrete Anwendung gemacht wurden. Insbesondere wird dabei auf die Vorteile und die Schwächen von MASIF eingegangen, die sich im Zuge der Umsetzung der Spezifikation gezeigt haben.

4.1 MASIF als vereinheitlichte Nomenklatur

Die größten Vorteile von MASIF sind die breite Unterstützung durch das Firmenkonsortium, das die Object Management Group (OMG) bildet, und die Tatsache, dass die MASIF–Spezifikation eine technologieübergreifende und implementierungsunabhängige Spezifikation und nicht eine proprietäre Lösung darstellt. Mit MASIF wird eine einheitliche Nomenklatur eingeführt. Es wird die Semantik der grundlegenden Begriffe (vgl. Abschnitt 2) definiert. MASIF hat das Konzept des Agentensystems als Ausführungs– und Ablaufumgebung für Mobile Agenten eingeführt. Innerhalb der MASIF–Spezifikation erfolgt auch die zentrale Registrierung für MASIF–konforme Agentensystemtypen. Dies sind die grundlegenden Schritte, um eine Interoperabilität von verschiedenen Agentensystemtypen überhaupt erreichen zu können.

Daneben wurden mit Regions und Places zwei Möglichkeiten zur Domänenbildung geschaffen. Dies ist insbesondere für Managementanwendungen, die organisatorische Grenzen überschreiten, von großer Bedeutung, also für Beziehungen wie Business–to–Administration (B2A), Business–to–Business (B2B), Business–to–Consumer (B2C), für das Outsourcing von Applikationen (Application Service Provider, ASP) und Prozessen (Business Process Outsourcing, BPO). Alle Agentensysteme, für die eine bestimmte Organisation verantwortlich ist, werden zu einer Region zusammengefasst. Aber auch innerhalb eines Agentensystems lassen sich Domänen bilden. Verschiedene Agenten von verschiedenen Authorities, die auf dem selben Agentensystem ausgeführt werden, können durch das Konzept der Places voneinander getrennt werden.

MASIF definiert ein einheitliches Namensschema sowohl für Agenten als auch Agentensysteme. Eine einheitliche und konsistente Benennung ist die Grundlage für die Lokalisierung von Agenten und Agentensystemen. Um die handelnden Subjekte, sowie die für die durchgeführten Aktionen Verantwortlichen überhaupt bestimmen, identifizieren bzw. authentisieren zu können, ist eine zweifelsfreie Benennung der Komponenten erforderlich, die im Auftrag eines Verantwortlichen handeln.

Neben der Nomenklatur wird auch das Management der Mobilen Agenten selbst und ein Satz von Basisschnittstellen festgelegt, die ein Agentensystem zur Verfügung stellen muss. Damit wird es möglich, dass ein Administrator Mobile Agenten starten, migrieren, terminieren, suspendieren, reaktivieren und kontaktieren, d.h. benutzen und verwalten kann, ohne genauere Kenntnisse des tatsächlich verwendeten Typs des Agentensystems oder des Mobilen Agenten zu besitzen. Für diese in MASIF spezifizierten Methoden werden auch Schnittstellendefinitionen in Form einer IDL–Spezifikation angegeben. Dies hat einerseits den Vorteil, dass alle MASIF–konformen Agentensysteme den gleichen Satz von Aufrufschnittstellen zur Verfügung stellen, und zwar unabhängig von der verwendeten Implementierungssprache. Andererseits können die IDL–Definitionen verwendet werden, um mit Hilfe eines IDL–Compilers Code–Rahmen für die entsprechenden Methoden zu erzeugen. Für die Implementierung bietet IDL den Vorteil, dass Datentypen und –formate, die in den in IDL beschriebenen Signaturen von Methoden verwendet werden, sprachübergreifend gültig bzw. identisch sind.

4.2 MASIF als kleinster gemeinsamer Nenner

Neben den Stärken, die MASIF konzeptionell bietet, sind insbesondere, wenn ein MASIF–konformes System implementiert wird, auch Nachteile und Schwächen in der Spezifikation zu erkennen. Das Hauptproblem bei der Implementierung ist, dass MASIF nur die Schnittstellen des Agentensystems spezifiziert und die Mobilen Agenten selbst überhaupt nicht betrachtet. MASIF spezifiziert keinerlei Interfaces von Agenten. Daher sind auch keine Informationen bezüglich der Inter–Agentenkommunikation in MASIF zu finden. Es bleibt dem Implementierer überlassen, wie Interfaces und Kommunikation von Agenten technisch realisiert werden. MASIF spezifiziert die Interoperabilität auf dem „kleinsten gemeinsamen Nenner“, nämlich den Schnittstellen des Agentensystems. Eine volle Interoperabilität, von Agenten und Agentensystemen verschiedener Hersteller ist dadurch nicht erreichbar. Die Spezifikation gibt keinerlei Hinweise wie z.B. erreicht werden kann, dass der Agent eines Herstellers (z.B. Aglets von IBM [OKO 98]) auf ein Agentensystem eines anderen Herstellers (z.B. MASA) migrieren und dort auch ausgeführt werden kann. Es werden auch keine Implementierungshinweise bezüglich Laufzeitumgebung, Serialisierung oder Implementierung der Migration gegeben. Bei dem Versuch, eine transparente Migration von verschiedenen Mobilen Agenten auf das gleiche Agentensystem zu implementieren, wird dieser Mangel der Spezifikation besonders deutlich. In [Bran 99] wurde z.B. versucht Voyager Agenten in MASA zu integrieren und, obwohl die Voyager Agenten ebenfalls in Java implementiert sind und eine CORBA–Schnittstelle bieten, sind die Basiskonzepte so unterschiedlich, dass eine direkte Integration und damit eine volle Interoperabilität nicht realisierbar ist. Es musste ein eigenes Gateway innerhalb von MASA implementiert werden, um die Unterschiede in den beiden Agententechnologien aufeinander abzubilden.

Obwohl MASIF von der OMG spezifiziert wurde und CORBA als Basis vorausgesetzt wird, fehlen die Bezüge zur Object Management Architecture [OMG 00-06-41] und zu anderen CORBA–Services, die von einem System Mobiler Agenten sinnvoll genutzt werden könnten. Zum Teil werden sogar eigene Konzepte entwickelt anstatt schon vorhandene Dienste zu nutzen. MASIF spezifiziert zur Lokalisierung und als Verzeichnisdienst für Mobile Agenten den MAFFinder. CORBA spezifiziert mit dem Naming Service einen eigenen Verzeichnisdienst, der von MASIF überhaupt nicht betrachtet wird. Geht man davon aus, dass Mobile Agenten als CORBA Objekte realisiert werden, so könnte der Naming Service als Verzeichnisdienst sehr einfach genutzt werden. Diese möglichen Synergieeffekte werden von MASIF jedoch nicht berücksichtigt.

5 Zusammenfassung und Ausblick

Mobile Agenten zeigen einen Weg auf, mit dem im Bereich des Managements von heterogenen IT–Systemen der sich stellende Bedarf an flexiblen und anpassbaren Lösungen decken ließe. Die MASIF–Spezifikation leistet dabei den Beitrag einer Konvergenz von unterschiedlichen Agentensystemen auf Basis des kleinsten gemeinsamen Nenners. Anhand des MASA–Agentensystems wird gezeigt, dass sich Mobile Agenten für das IT–Management einsetzen lassen. Dabei hat sich aber auch gezeigt, dass MASIF alleine für eine nahtlose Integration verschiedener Agentensysteme in der Praxis bei weitem nicht ausreicht.

Um Mobile Agenten zu einer breiten Akzeptanz verhelfen zu können, sind jedoch noch weitere Forschungen im Bereich der Sicherheit notwendig, der sich das MNM–Team u.a. widmet. Weiterhin wird im Rahmen des BMBF–Forschungsprojekts "Naturanaloge Lern– und Optimierungsverfahren für vernetzte Systeme, LEONET -KM-" in Kooperation mit der Siemens AG, München, MASA als Management System eingesetzt. In dieser Kooperation, wird der in [Ense 99] vorgestellte Ansatz, basierend auf MASA umgesetzt und dabei auch der Einsatz von KI–Methoden in Form neuronaler Netze zu Managementzwecken untersucht.

Danksagung

Die Autoren danken dem Münchner Netzmanagement Team für intensive Diskussionen zu früheren Versionen dieses Beitrages. Das MNM–Team, das von Prof. Hegering geleitet wird, ist ein Gruppe von Wissenschaftlern beider Münchner Universitäten und des Leibniz–Rechenzentrums der Bayerischen Akademie der Wissenschaften. Der Webserver des Teams ist unter `http://wwwmnmteam.informatik.uni-muenchen.de` zu fin-

den.

Literatur

[BPW 98] Bieszczad, A., B. Pagurek und T. White: *Mobile Agents for Network Management.* IEEE Communication Surveys, 1(1), 1998, `http://dsp.jpl.nasa.gov/members/payman/swarm/bieszczad98-comsoc.pdf` .

[Bran 99] Brandt, R.: *Interoperabilität von CORBA ORBs in Mobilen Agentensystemen.* Systementwicklungsprojekt, Technische Universität München, Mai 1999, `http://wwwmnmteam.informatik.uni-muenchen.de/common/Literatur/MNMPub/Fopras/bran99/bran99.shtml` .

[CORBA 2.2] *The Common Object Request Broker: Architecture and Specification.* OMG Specification Revision 2.2, Object Management Group, Februar 1998.

[Demm 99] Demmel, S.: *Design und prototypische Implementierung eines policy-basierten Dienstmanagements für das Nomadic Computing.* Diplomarbeit, Ludwig-Maximilians-Universität München, November 1999.

[Ense 99] Ensel, C.: *Automated generation of dependency models for service management.* In: *Workshop of the OpenView University Association (OVUA'99)*, Bologna, Italien, Juni 1999. .

[FIPA 00] Foundation for Intelligent Physical Agents: *Abstract Architecture Specification*, November 2000, `http://www.fipa.org/specs/fipa00001/` .

[FIPA 98] Foundation for Intelligent Physical Agents: *FIPA 98 Specification*, 1998, `http://www.fipa.org` .

[Funf 98] Fünfrocken, S.: *Transparent Migration of Java–Based Mobile Agents.* In: Rothermel, K. und F. Hohl (Herausgeber): *Mobile Agents (MA '98)*, Band 1477 der Reihe *LNCS*, Berlin; Heidelberg, 1998. Springer.

[Gerb 99] Gerber, S.: *Entwicklung einer Benutzerschnittstelle mit Java/CORBA für die Mobile Agent System Architecture (MASA).* Systementwicklungsprojekt, Technische Universität München, April 1999, `http://wwwmnmteam.informatik.uni-muenchen.de/common/Literatur/MNMPub/Fopras/gerb99/gerb99.shtml` .

[Gigl 00] Gigl, J. G.: *Implementierung der MAFFinder–Schnittstelle für die Mobile Agent System Architecture.* Studienarbeit für das Aufbaustudium Informatik, Technische Universität München, Juli 2000, `http://www.nm.informatik.uni-muenchen.de/common/Literatur/MNMPub/Fopras/gigl00/gigl00.shtml` .

[Gold 96] Goldszmidt, Germán S.: *Distributed Management by Delegation.* Dissertation, Graduate School of Arts and Sciences, Columbia University, 1996.

[Grus 99] Gruschke, B.: *Entwurf eines Eventkorrelators mit Abhängigkeitsgraphen.* Dissertation, Ludwig-Maximilians-Universität München, November 1999.

[HAN 99a] Hegering, H.-G., S. Abeck und B. Neumair: *Integriertes Management vernetzter Systeme — Konzepte, Architekturen und deren betrieblicher Einsatz.* dpunkt-Verlag, ISBN 3-932588-16-9, 1999, `http://www.dpunkt.de/produkte/management.html` .

[HaRe 00] Hauck, R. und H. Reiser: *Monitoring Quality of Service across Organizational Boundaries.* In: Linnhoff-Popien, C. und H.-G. Hegering (Herausgeber): *Proceedings of the 3rd IFIP/GI International Conference on Trends towards a Universal Service Market (USM 2000)*, Nummer 1890 in *Lecture Notes in Computer Science*, Munich, Germany, September 2000. Springer.

[Heil 99] Heilbronner, S.: *Konzeption einer Architektur für das integrierte Management der Ressourcennutzung nomadischer Systeme in Datennetzen.* Dissertation, Ludwig-Maximilians-Universität München, Dezember 1999.

[Hojn 99] Hojnacki, M.: *Einsatz des Java Dynamic Management Kit (JDMK) zur Antwortzeitüberwachung bei der DeTeSystem* . Diplomarbeit, Technische Universität München, Mai 1999, `http://wwwmnmteam.informatik.uni-muenchen.de/common/Literatur/MNMPub/Diplomarbeiten/hojn99/hojn99.shtml` .

[Kell 98] Keller, A.: *CORBA-basiertes Enterprise Management: Interoperabilität und Managementinstrumentierung verteilter kooperativer Managementsysteme in heterogener Umgebung.* Dissertation, Technische Universität München, Dezember 1998.

[Kemp 98] Kempter, B.: *Entwurf eines Java/CORBA-basierten Mobilen Agenten.* Diplomarbeit, Technische Universität München, August 1998.

[Knoe 99] Knöchlein, H.: *Management eines Internet Telefonie Servers mittels JDMK.* Diplomarbeit, Technische Universität München, Februar 1999, http://wwwmnmteam.informatik.uni-muenchen.de/common/Literatur/MNMPub/Diplomarbeiten/knoe99/knoe99.shtml .

[KQML 97] Labrou, Y. und T. Finin: *A Proposal for a new KQML Specification.* Technical Report TR CS–97–03, University of Maryland Baltimore County, Baltimore, Februar 1997, http://www.csee.umbc.edu/kqml/papers/ .

[LeSc 99] Levi, D. und J. Schoenwaelder: *RFC 2592: Definitions of Managed Objects for the Delegation of Management Script.* RFC, IETF, Mai 1999, ftp://ftp.isi.edu/in-notes/rfc2592.txt .

[MASIF] *Mobile Agent System Interoperability Facilities Specification.* OMG TC Document orbos/98-03-09, Object Management Group, März 1998.

[Moun 97] Mountzia, M.-A.: *Flexible Agents in Integrated Network and Systems Management.* Dissertation, Technische Universität München, Dezember 1997.

[OKO 98] Oshima, M., G. Karjoth und K. Ono: *Aglets Specification 1.1 Draft.* Technischer Bericht, IBM Research Labs Tokio, September 1998, http://www.trl.ibm.co.jp/aglets/spec11.html .

[OMG 00-06-41] *Discussion of the Object Management Architecture (OMA) Guide.* OMG Specification formal/00-06-41, Object Management Group, Juni 2000, ftp://ftp.omg.org/pub/docs/formal/00-06-41.pdf .

[Radi 98] Radisic, I.: *Konzeption eines policy-basierten Konfigurationsmanagements für nomadische Systeme in Intranets.* Diplomarbeit, Ludwig-Maximilians-Universität München, August 1998, http://www.nm.informatik.uni-muenchen.de/common/Literatur/MNMPub/Diplomarbeiten/radi98/radi98.shtml .

[Reis 99] Reiser, H.: *Einsatz des Java Dynamic Management Kit für flexible, dynamische Managementsysteme.* PIK – Praxis der Informationsverarbeitung und Kommunikation, 22(4):204–212, Dezember 1999, http://wwwmnmteam.informatik.uni-muenchen.de/common/Literatur/MNMPub/Publikationen/reis99/reis99.shtml .

[ReVo 00a] Reiser, H. und G. Vogt: *Security Requirements for Management Systems using Mobile Agents.* In: Tohme, S. und M. Ulema (Herausgeber): *Proceedings of the Fifth IEEE Symposium on Computers & Communications*, Seiten 160–165, Antibes - Juan Les Pins, France, Juli 2000. IEEE, http://wwwmnmteam.informatik.uni-muenchen.de/common/Literatur/MNMPub/Publikationen/revo00a/revo00a.shtml .

[RFC 1157] Case, J. D., M. Fedor, M. L. Schoffstall und C. Davin: *RFC 1157: Simple Network Management Protocol (SNMP).* RFC, IETF, Mai 1990, http://sunsite.auc.dk/RFC/rfc/rfc1157.html .

[Roel 99a] Rölle, H.: *Authentisierung und Autorisierung für das Java/CORBA–Agentensystem MASA.* Diplomarbeit, Technische Universität München, August 1999, http://www.nm.informatik.uni-muenchen.de/common/Literatur/MNMPub/Diplomarbeiten/roel99/roel99.shtml .

[Roel 99b] Rölle, H.: *Prototypische Implementierung des CORBA Topology Service.* Fortgeschrittenenpraktikum, Technische Universität München, Februar 1999.

[Schö 97] Schönwälder, J.: *Network Management by Delegation - From Research Prototypes Towards Standards.* In: *8th Joint European Networking Conf. (JENC8)*, Edinburgh, Mai 1997. .

[Voyager 2.0] Objectspace: *Voyager Core Technology 2.0 – Users Guide.* Technischer Bericht, Objectspace, 1998.