

***Nachrichtenorientierte Verteilungsplattform für hochdynamische
Kommunikation in der Produktion***

Von der Fakultät für Konstruktions-, Produktions- und Fahrzeugtechnik
der Universität Stuttgart zur Erlangung der Würde eines
Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

Vorgelegt von

Dipl.-Inf. Gerald Knoll

aus Aalen

Hauptberichter: Univ.-Prof. Dr.-Ing. Prof. E.h. Dr.-Ing. E.h. Dr. h.c.
mult. E. Westkämper
Mitberichter: Univ.-Prof. Dr.-Ing. Dr.-Ing. E.h. Dieter Spath

Tag der Einreichung: 18.12.2007
Tag der mündlichen Prüfung: 18.11.2008

IPA-IAO Forschung und Praxis

Berichte aus dem
Fraunhofer-Institut für Produktionstechnik und
Automatisierung (IPA), Stuttgart,
Fraunhofer-Institut für Arbeitswirtschaft und
Organisation (IAO), Stuttgart,
Institut für Industrielle Fertigung und
Fabrikbetrieb (IFF), Universität Stuttgart
und Institut für Arbeitswissenschaft und
Technologiemanagement (IAT), Universität Stuttgart


Herausgeber:

Univ.-Prof. Dr.-Ing. Prof. e.h. Dr.-Ing. e.h. Dr. h.c. mult. Engelbert Westkämper
und

Univ.-Prof. Dr.-Ing. habil. Prof. E.h. mult. Dr. h.c. mult. Hans-Jörg Bullinger
und

Univ.-Prof. Dr.-Ing. Dr.-Ing. E.h. Dieter Spath

Gerald Knoll



Nachrichtenorientierte Verteilungsplattform für hochdynamische Kommunikation in der Produktion

Nr. 486

Dr.-Ing. Dipl.-Inf. Gerald Knoll

Fraunhofer Institut für Produktionstechnik und Automatisierung (IPA), Stuttgart

Univ.-Prof. Dr.-Ing. Prof. e.h. Dr.-Ing. e.h. Dr. h.c. mult. Engelbert Westkämper

ord. Professor an der Universität Stuttgart

Fraunhofer-Institut für Produktionstechnik und Automatisierung (IPA), Stuttgart

Univ.-Prof. Dr.-Ing. habil. Prof. E.h. mult. Dr. h.c. mult. Hans-Jörg Bullinger

ord. Professor an der Universität Stuttgart

Präsident der Fraunhofer-Gesellschaft, München

Univ.-Prof. Dr.-Ing. Dr.-Ing. E.h. Dieter Spath

ord. Professor an der Universität Stuttgart

Fraunhofer-Institut für Arbeitswirtschaft und Organisation (IAO), Stuttgart

D 93

ISBN (10) 3-939890-47-2, ISBN (13) 978-3-939890-47-8

Jost Jetter Verlag, Heimsheim

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils gültigen Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

© Jost-Jetter Verlag, Heimsheim 2009.

Printed in Germany.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Sollte in diesem Werk direkt oder indirekt auf Gesetze, Vorschriften oder Richtlinien (z. B. DIN, VDI, VDE) Bezug genommen oder aus ihnen zitiert worden sein, so kann der Verlag keine Gewähr für die Richtigkeit, Vollständigkeit oder Aktualität übernehmen. Es empfiehlt sich, gegebenenfalls für die eigenen Arbeiten die vollständigen Vorschriften oder Richtlinien in der jeweils gültigen Fassung hinzuzuziehen.

Druck: printsystem GmbH, Heimsheim

Geleitwort der Herausgeber

Über den Erfolg und das Bestehen von Unternehmen in einer marktwirtschaftlichen Ordnung entscheidet letztendlich der Absatzmarkt. Das bedeutet, möglichst frühzeitig absatzmarktorientierte Anforderungen sowie deren Veränderungen zu erkennen und darauf zu reagieren.

Neue Technologien und Werkstoffe ermöglichen neue Produkte und eröffnen neue Märkte. Die neuen Produktions- und Informationstechnologien verwandeln signifikant und nachhaltig unsere industrielle Arbeitswelt. Politische und gesellschaftliche Veränderungen signalisieren und begleiten dabei einen Wertewandel, der auch in unseren Industriebetrieben deutlichen Niederschlag findet.

Die Aufgaben des Produktionsmanagements sind vielfältiger und anspruchsvoller geworden. Die Integration des europäischen Marktes, die Globalisierung vieler Industrien, die zunehmende Innovationsgeschwindigkeit, die Entwicklung zur Freizeitgesellschaft und die übergreifenden ökologischen und sozialen Probleme, zu deren Lösung die Wirtschaft ihren Beitrag leisten muss, erfordern von den Führungskräften erweiterte Perspektiven und Antworten, die über den Fokus traditionellen Produktionsmanagements deutlich hinausgehen.

Neue Formen der Arbeitsorganisation im indirekten und direkten Bereich sind heute schon feste Bestandteile innovativer Unternehmen. Die Entkopplung der Arbeitszeit von der Betriebszeit, integrierte Planungsansätze sowie der Aufbau dezentraler Strukturen sind nur einige der Konzepte, welche die aktuellen Entwicklungsrichtungen kennzeichnen. Erfreulich ist der Trend, immer mehr den Menschen in den Mittelpunkt der Arbeitsgestaltung zu stellen - die traditionell eher technokratisch akzentuierten Ansätze weichen einer stärkeren Human- und Organisationsorientierung. Qualifizierungsprogramme, Training und andere Formen der Mitarbeiterentwicklung gewinnen als Differenzierungsmerkmal und als Zukunftsinvestition in *Human Resources* an strategischer Bedeutung.

Von wissenschaftlicher Seite muss dieses Bemühen durch die Entwicklung von Methoden und Vorgehensweisen zur systematischen Analyse und Verbesserung des Systems Produktionsbetrieb einschließlich der erforderlichen Dienstleistungsfunktionen unterstützt werden. Die Ingenieure sind hier gefordert, in enger Zusammenarbeit mit anderen Disziplinen, z. B. der Informatik, der Wirtschaftswissenschaften und der Arbeitswissenschaft, Lösungen zu erarbeiten, die den veränderten Randbedingungen Rechnung tragen.

Die von den Herausgebern langjährig geleiteten Institute, das

- Fraunhofer-Institut für Produktionstechnik und Automatisierung (IPA),
- Fraunhofer-Institut für Arbeitswirtschaft und Organisation (IAO),
- Institut für Industrielle Fertigung und Fabrikbetrieb (IFF), Universität Stuttgart,
- Institut für Arbeitswissenschaft und Technologiemanagement (IAT), Universität Stuttgart

arbeiten in grundlegender und angewandter Forschung intensiv an den oben aufgezeigten Entwicklungen mit. Die Ausstattung der Labors und die Qualifikation der Mitarbeiter haben bereits in der Vergangenheit zu Forschungsergebnissen geführt, die für die Praxis von großem Wert waren. Zur Umsetzung gewonnener Erkenntnisse wird die Schriftenreihe „IPA-IAO - Forschung und Praxis“ herausgegeben. Der vorliegende Band setzt diese Reihe fort. Eine Übersicht über bisher erschienene Titel wird am Schluss dieses Buches gegeben.

Dem Verfasser sei für die geleistete Arbeit gedankt, dem Jost Jetter Verlag für die Aufnahme dieser Schriftenreihe in seine Angebotspalette und der Druckerei für saubere und zügige Ausführung. Möge das Buch von der Fachwelt gut aufgenommen werden.

Engelbert Westkämper Hans-Jörg Bullinger Dieter Spath

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter der Abteilung Reinst- und Mikroproduktion am Fraunhofer Institut für Produktionstechnik und Automatisierung IPA, Stuttgart.

Herrn Professor Dr.-Ing. Prof. E.h. Dr.-Ing. E.h. Dr. h.c. mult. Engelbert Westkämper danke ich für die wissenschaftliche Betreuung und Förderung der Arbeit sowie die Übernahme des Hauptberichts. In gleicher Weise danke ich Herrn Professor Dr.-Ing. Dr.-Ing. E.h. Dieter Spath für die Übernahme des Mitberichts sowie die eingehende Durchsicht meiner Arbeit. Meinen Dank möchte ich ebenso Herrn Professor Dr.-Ing. Dr. h.c. mult. Rolf Dieter Schraft, Dr.-Ing. Johann Dorner und Joachim Seidelmann für die wohlwollende Unterstützung und Ermöglichung dieser Arbeit aussprechen.

Eine besondere Anerkennung gebührt allen beteiligten Kolleginnen und Kollegen der Abteilung Reinst- und Mikroproduktion für die zielführenden Diskussionen und die konstruktive Kritik während der Entstehung meiner Arbeit. Explizit möchte ich mich bei den Herren Dr.-Ing. Ralf Muckenhirn und Dr.-Ing. Akhauri Kumar für die wertvolle Unterstützung bei der Herausarbeitung von Themenschwerpunkten sowie bei Jutta Knoll, Dr.-Ing. Udo Gommel, Dr.-Ing. Philipp Dreiss, Matthias Meier, Michael Hoffmeister, Dr.-Ing. Frank Frauenhoffer und Christian Fischmann für die Unterstützung bei der Strukturierung der Arbeit.

Zum Dank für das entgegengebrachte Verständnis, das Vertrauen und den großen Rückhalt während des Entstehens der Arbeit widme ich das Buch meiner Frau Jutta Knoll und meinen Eltern Lilli und Rudi Knoll.

Stuttgart, im Mai 2009

Gerald Knoll

Inhaltsverzeichnis

1	Einleitung	15
1.1	Problemstellung und Herausforderung	15
1.2	Zielsetzung und Vorgehensweise	17
2	Ausgangssituation und Analyse	19
3	Stand der Technik	27
3.1	Kommunikation	27
3.2	Verteilungsplattform	30
3.2.1	Marktanalyse über Verteilungsplattformen	33
3.2.2	Bewertung von Verteilungsplattformen	35
3.3	Nachrichtenorientierte Verteilungsplattform	37
3.3.1	Verteilte Nachrichtenorientierte Verteilungsplattform	41
3.3.2	Bewertung von Nachrichtenorientierten Verteilungsplattformen	43
3.4	Kommunikationsmodelle einer Nachrichtenorientierten Verteilungs- plattform	45
3.4.1	Bewertung der Kommunikationsmodelle	48
3.5	Namensdienst	49
3.5.1	Bewertung des Namensdienstes	51
3.6	Routenplanung	52
3.6.1	Bewertung der Routenplanung	54
3.7	Arbeitsablaufverwaltung	55
3.7.1	Bewertung der Arbeitsablaufverwaltung	56
3.8	Nachrichtenzustellung in der Wartung	58
3.8.1	Bewertung der Nachrichtenzustellung in der Wartung	59
3.9	Zusammenfassung der Anforderungen	60
3.10	Bewertung der abgeleiteten Anforderungen einer Nachrichtenorientierten Verteilungsplattform für die Produktion	61
4	Konzeption und Entwicklung	65
4.1	Konzeption	65
4.1.1	Nachrichtenübermittlung	66
4.1.2	Namensdienst	71
4.1.3	Routenplanung	72
4.1.4	Arbeitsablaufverwaltung	73
4.2	Entwicklung des Namensdienstes	73
4.3	Entwicklung der Routenplanung	77
4.4	Entwicklung der Arbeitsablaufabstimmung	81
4.5	Integration der Komponenten in das Gesamtsystem	88
5	Realisierung und Anwendung	93
5.1	Namensdienst	94
5.2	Routenplanung	96
5.3	Arbeitsablaufabstimmung	97
5.4	Anwendung der integrierten Nachrichtenorientierten Verteilungsplattform ..	98
6	Verifikation und Bewertung	101

6.1	Punkt-zu-Punkt Nachrichtenübermittlung mit höherem Durchsatz als bei bestehenden Systemen	101
6.2	Abbildung von komplexen Beziehungen im Namensraum.....	112
6.3	Einbeziehung aller Informationen des Empfängers für eine situationsspezifische Anpassung der Nachricht an das optimale Nachrichtenprotokoll des Empfängers.....	112
6.4	Ausführung von Arbeitsabläufen von Sender und Empfänger unter Berücksichtigung der Situation des Gesamtsystems	113
7	Zusammenfassung und Ausblick	115
7.1	Grundlagen zur Weiterentwicklung.....	117
8	Summary	119
9	Literatur.....	125
10	Anhänge.....	153
10.1	Anhang A - Namensdienste	153
10.2	Anhang B - Nachrichtenorientierte Verteilungsplattformen	153

Abkürzungen und Formelzeichen

24/7	24 Stunden am Tag / 7 Tage die Woche
A(x)	Datenstruktur
ADO	ActiveX Data Object
API	Application Programming Interface
ASI	Aktuator Sensor Interface
AVI	Audio Video Interleaved
BPE	Business Process Engine
BPM	Business Process Modelling
BZ	Bewertungszahl
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CAN	Controller Area Network
CC / PP	Composite Capability / Preference Profiles
COM	Component Object Model
CPU	Central Processing Unit
DB	Datenbank
DCOM	Distributed Component Object Model
DNS	Internet Domain Name System
E	Menge der Kanten
EAI	Enterprise Application Integration
EJB	Enterprise Java Beans
ERP	Enterprise Resource Planning
ESB	Enterprise Service Bus
FIB	Factory Implementation Protocol
G	Graph
GEM	Generic Equipment Model
GHz	Gigahertz
GIF	Graphics Interchange Format
GNS	Global Name Service
GSS	Generic Security Services
HTML	Hypertext Markup Language

HTTP	Hypertext Transfer Protocol
INS	Interoperable Naming Service
ISO	Internationale Organisation für Normung
JAAS	Java Authentication and Authorization Service
JCA	J2EE Connection Adapter
JCE	Java Cryptography Extension
JDBC	Java Data Base Connector
JDO	Java Data Objects
JMS	Java Messaging Service
JNDI	Java Naming and Directory Interface
JPEG	Joint Photographic Experts Group
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LON	Local Operating Network
M2A	Message to Anywhere
MBean	Managed Bean
MBit	Megabit
MOM	Message Oriented Middleware
MPEG	Moving Picture Experts Group
MS	Microsoft
O()	Laufzeitverhalten
OOM	Object Oriented Middleware
OPC	OLE for Process Control
ORB	Object Request Brokers
OSI	Open Systems Interconnection
PDA	Personal Digital Assistant
PROFIBUS	Process Field Bus
PPS	Produktionsplanung und -steuerung
PSS	Produktionssteuerungssystem
RAM	Random Access Memory
RfE	Request for Evaluation
RFID	Radio Frequency Identification
RMI	Remote Method Invocation

RPC	Remote Procedure Call
RS-232	Recommended Standard 232, Serielle Schnittstelle
SECS	SEMI Equipment Communication Standard
SME	Small Medium Enterprise
SMS	Short Messaging Service
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
TPM	Transaction Processing Monitors
UML	Unified Modelling Language
V	Menge der Knoten
VDE-IT	Verband der Elektrotechnik, Elektronik und Informationstechnik
VDI	Verein Deutscher Ingenieure
WfM	Workflow-Management
WLAN	Wireless LAN
WWS	Warenwirtschaftssystem
XML	Extensible Markup Language
Ψ	Abbildung

1 Einleitung

1.1 *Problemstellung und Herausforderung*

Durch immer größere Produktionsstätten mit einer wachsenden Anzahl von Anlagen und einem steigenden Grad der **Automatisierung** wird die **Produktion** immer **komplexer** [Dorf94, ITRS04, West06]. Eine steigende Automatisierung bis hin zur Vollautomatisierung bedeutet, dass alle Anlagen, Geräte, Applikationen, Softwaresysteme, Datenbanken, PCs und mobilen Geräte innerhalb der Produktion miteinander kommunizieren müssen. Durch neue Formen von Unternehmensnetzwerken, zunehmender Modularisierung und arbeitsteiliger Organisation hängt der wirtschaftliche Erfolg einer Fertigung in entscheidendem Maße vom Austausch von Informationen ab [Stöckel01]. Die richtige Information, am richtigen Ort, zum richtigen Zeitpunkt, in der richtigen Darstellung wird daher zu einem integrierenden und einem der wichtigsten Produktionsfaktoren [Weule92, Schmid93]. In einem Unternehmen mit einer verteilten Produktionsumgebung sind die Verfügbarkeit und der Zugriff auf aktuelle Prozessdaten unerlässlich, da auf deren Grundlage fundierte Entscheidungen getroffen werden [Feldmann98]. Die Anwendungen in der Produktion sind die Produktionssteuerung, betriebsbegleitende Planung und Simulation, Fernwartung von Anlagen sowie die Überwachung von Produktionsprozessen in der Leitebene [Göhringer99]. Heutige Märkte fordern maßgeschneiderte hochqualitative Produkte mit kürzeren Lebenszyklen, die nur durch die Erhöhung der Flexibilität, Rekonfigurierbarkeit und Reaktionsschnelligkeit der Fertigung und deren Kommunikationstechnologien erreicht werden können [Desfo06, Picot96]. Bei der immer wichtiger werdenden Anforderung nach einer Multivarianten-Fertigung mit steigendem Produktmix wird eine höhere Bandbreite und Dynamik in der Kommunikation benötigt [Mertens94, Arnold95], da schnell auf eine geänderte Situation in der Produktion reagiert werden muss.

Eine **Nachrichtenorientierte Verteilungsplattform** (engl. Message Oriented Middleware, Abk. MOM) [Emm00, Ha05] sorgt dafür, dass die Kommunikation zwischen allen Anlagen, Applikationen und Systemen aufgebaut und standardisiert werden kann und wird deshalb innerhalb der Produktion und Automatisierung immer

wichtiger. In der Produktion gibt es noch keinen in sich geschlossenen Ansatz, eine Verteilungsplattform zu entwerfen, da es keine Standardmethode gibt, Kommunikation aufzubauen [Werth03]. Thema dieser Arbeit ist ein Gesamtkonzept für den Aufbau und die Konfiguration von Basis-Komponenten einer Nachrichtenorientierten Verteilungsplattform, um den wachsenden Anforderungen der Produktion hinsichtlich der Nachrichtenübertragungsgeschwindigkeit und der Dynamik gerecht zu werden.

Als Produktion wird in dieser Arbeit die Massenfertigung mit einer hohen Anzahl von Anlagen betrachtet. In der **Halbleiterproduktion** sind beispielsweise in einer Produktionshalle einer 300mm Fertigung [Schulz00] etwa 600 Anlagen und 1000 Geräte, Applikationen, PCs und Systeme vorhanden, die im Sekundentakt Nachrichten austauschen müssen [Osborne06, Mönch03, M2A03a]. Somit kann es auf einen Nachrichtendurchsatz von mehreren tausend Nachrichten in der Sekunde kommen, die versendet und verarbeitet werden müssen, bei gleichzeitig hoher Dienstgüte wie z.B. fehlerfreier und sicherer Zustellung.

Durch die stetig steigende Anzahl von Nachrichten in der Produktion und durch immer komplexere Nachrichten, wie z.B. detaillierte Sensordaten, die von Systemen benötigt werden [Heide01, Intana00] oder durch neue Standardisierungen in der Kommunikation von Anlagen, wie z.B. XML, wodurch viele semantische Informationen gesendet werden müssen [Sutter05, Dreiss05, Meier06], entsteht die Herausforderung einer hohen und ständig steigenden **Kommunikationsdichte**.

Nur wenige am Markt verfügbare Verteilungsplattformen können die steigende Kommunikationsdichte einer hochautomatisierten Produktion mit mehreren tausend Nachrichten pro Sekunde bewältigen. Da die Bandbreite bei der Kommunikation nicht beliebig erhöht werden kann und man dabei von neuen Netzwerk-Technologien abhängig ist, besteht die Anforderung an eine Verteilungsplattform für eine **schnellere Nachrichtenübermittlung**.

Parallel zu einer schnellen Nachrichtenübermittlung treten innerhalb von Applikationen in der Produktion wie beispielsweise einem **Wartungssystem** Anforderungen an eine **flexible und hochdynamische Nachrichtenübermittlung** auf [Pend03].

1.2 Zielsetzung und Vorgehensweise

In dieser Arbeit werden **Basis-Komponenten** für eine Nachrichtenorientierte Verteilungsplattform entwickelt. Durch das Gesamtkonzept dieser Komponenten ist es möglich, bei einem Durchsatz von **mehreren tausend Nachrichten pro Sekunde** dennoch eine **flexible und hochdynamische Nachrichtenübermittlung** zu erreichen. Die Nachrichten werden dabei während der Zustellung situationspezifisch an den Empfänger angepasst.

Zuerst wird in dieser Arbeit auf den **Stand der Technik** der relevanten Teilgebiete zur Entwicklung einer Nachrichtenorientierten Verteilungsplattform eingegangen. Der Entwicklungsstand dieser Teilgebiete wird bewertet und daraus grundlegende Anforderungen an Nachrichtenorientierte Verteilungsplattformen in der Produktion abgeleitet.

Die **Konzeption und Entwicklung** von Komponenten einer hochdynamischen Nachrichtenorientierten Verteilungsplattform für die Produktion findet aufbauend auf den abgeleiteten Anforderungen statt.

Innerhalb der **Realisierung und Anwendung** wird eine prototypische Implementierung der Komponenten der Nachrichtenorientierten Verteilungsplattform aufgezeigt.

Eine **Bewertung** führt auf, dass die detaillierten Anforderungen durch die Komponenten der Nachrichtenorientierten Verteilungsplattform dieser Arbeit erfüllt werden.

Im Schlusskapitel folgen eine **Zusammenfassung** und ein **Ausblick** über weitere Entwicklungspotenziale innerhalb einer hochdynamischen Nachrichtenübermittlung in der Produktion.

2 Ausgangssituation und Analyse

Bei dem Versuch, ein Gesamtkonzept für eine schnelle Nachrichtenorientierte Verteilungsplattform für die Produktion zu entwickeln, die zudem eine hochdynamische Nachrichtenübermittlung unterstützt, stößt man auf Herausforderungen. Diese Herausforderungen werden im Folgenden analysiert.

Folgende Abbildung 2-1 zeigt eine Nachrichtenorientierte Verteilungsplattform in der Produktion, die eine Vielzahl von Anlagen, Systemen und Applikationen verbindet, die für die Produktion notwendig sind.

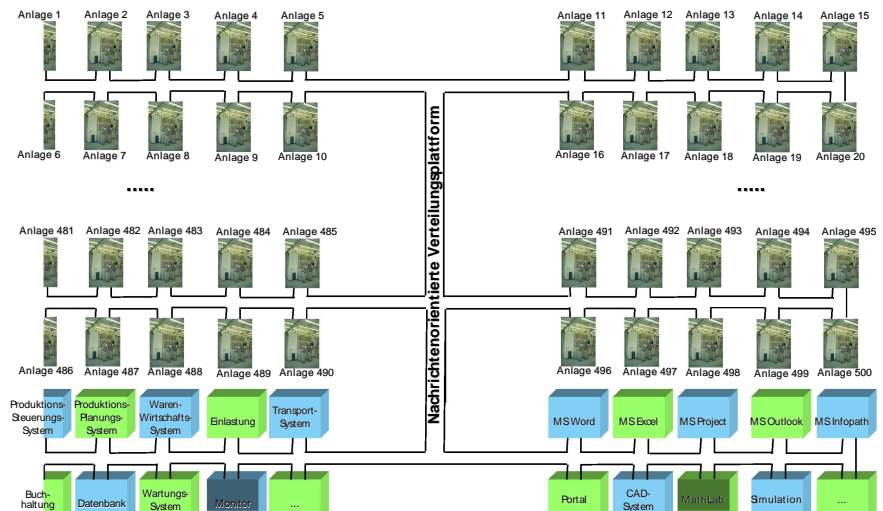


Abbildung 2-1: Nachrichtenorientierte Verteilungsplattform in der Produktion

Um Produktionsprozesse auszuführen, wird wie in Kapitel 1.1 beschrieben eine Kommunikation zwischen den Anlagen, Applikationen und Systemen benötigt, die bei der Massenproduktion zu einem Anstieg der **Kommunikationsdichte** führt. Dieser Anstieg kann verschiedene **Auswirkungen** auf die Produktion haben [IFX05]:

- Es treffen Nachrichten mit z.B. Anlagensteuerungsbefehlen verspätet ein und verzögern somit einen Produktionsprozess wodurch eine Anlage auf die Nach-

richt warten muss und in dieser Zeit nicht produzieren kann. Wenn diese Anlage auf einem kritischen Pfad in der Produktionslinie liegt, wird unter Umständen die gesamte Produktion verzögert.

- Die Verbindung zu einer Anlage kann ausfallen, wodurch keine Nachrichten mehr zugestellt werden können. Unter Umständen gehen die Nachrichten auf dem Weg verloren und die Produktion muss nach dem Wiederaufbau der Verbindung diese Nachrichten lokalisieren und nochmals versenden.
- Einzelne Applikationen wie z.B. die Überwachungskomponente oder Datenerfassung können nicht mehr benutzt werden, da sonst für den Produktionsprozess wichtige Nachrichten verspätet eintreffen und somit das Gesamtsystem instabil wird.
- Eine ständige Überlastung des Gesamtsystems führt zu hoher Fehleranfälligkeit, die den Absturz eines Servers und somit den Ausfall der Produktion mit sich führen kann.

Nur wenige am Markt verfügbare Verteilungsplattformen können die steigende Kommunikationsdichte einer hochautomatisierten Massenproduktion mit mehr als tausend Anlagen, Applikationen und Systemen bewältigen. Eine **schnellere Nachrichtenübermittlung** hilft, die oben dargestellten Auswirkungen zu vermeiden und verhilft der Produktion zu höheren Transaktionsraten bei der Nachrichtenübermittlung. Die steigenden Transaktionsraten führen zu einer schnelleren Verarbeitung von Daten und können aufgrund des höheren Informationsflusses zu einer höheren Qualitätssicherung in der Produktion beitragen [Ender03, Jacobsen04, Alonso05, Steen06]. Da die verfügbaren Nachrichtenorientierten Verteilungsplattformen, die die Anforderung einer schnellen Nachrichtenübermittlung erfüllen, gleichzeitig aber hinsichtlich Flexibilität und Dynamik den folgenden Herausforderungen nicht gerecht werden [Desfo06, Picot94, Stiedl04, Knoll06a], wird in dieser Arbeit ein Gesamtkonzept erstellt. Besonders durch das vielseitige Spektrum an Ein- und Ausgabegeräten, die in der Produktion verwendet werden und über verschiedene Netzwerke kommunizieren, gibt es einen Bedarf an Protokoll- und Inhaltsanpassungen zwischen den kommunizierenden Parteien [Jin04, Prasad04, Brom05]. Die Vielseitigkeit und Vielschichtigkeit der Transformationen benötigen einen großen Rechenaufwand [Mohan99, Chandra00] wenn sie nicht sogar unlösbar sind [Jin04].

Eine der Herausforderungen an eine flexible und hochdynamische Nachrichtenübermittlung in der Produktion tritt bei der Nachrichtenzustellung beispielsweise an ein Wartungssystem auf. Da die Wartung 15% bis 40% der Gesamtkosten in der Produktion ausmacht [Han06, Mobley90], muss eine Verteilungsplattform für die Produktion auch deren Anforderungen in punkto Flexibilität erfüllen. Durch das Auftreten von mobilen Geräten in der Wartung, die sich ständig von der Verteilungsplattform an- und abmelden oder den Zugangspunkt wechseln wird mehr Flexibilität und Dynamik benötigt, da sich alle Anlagen, Applikationen und Systeme ständig an ein sich änderndes Netzwerk anpassen müssen. In der Produktion bekommt ein Informationsverarbeiter durchschnittlich 51 Nachrichten am Tag an bis zu sieben unterschiedliche Orte zugestellt, auf die schnell und präzise reagiert werden muss [Muir07]. Da bei der Wartung auftretende Anlagenausfälle möglichst schnell behoben werden müssen, ist die Zustellung einer Nachricht an das Wartungspersonal von Bedeutung. Bei einem Ausfall einer Anlage in der Produktion ist es von Vorteil, wenn der am besten geeignete Wartungsarbeiter eine Nachricht über den Anlagenausfall erhält. Außerdem sollte dem Wartungsarbeiter die Nachricht automatisch auf genau das Eingabegerät zugestellt werden, mit dem er momentan am System angemeldet und verfügbar ist. Dadurch entsteht die Herausforderung, dass sich eine Nachricht an die Situation des Gesamtsystems dynamisch anpassen sollte. Dies führt zu der Anforderung einer

- **automatischen Umwandlung der Nachricht auf das optimale Empfangsgerät.**

Ein Beispiel für die Verwendung eines Wartungssystems in der Produktion ist in Abbildung 2-2 dargestellt.

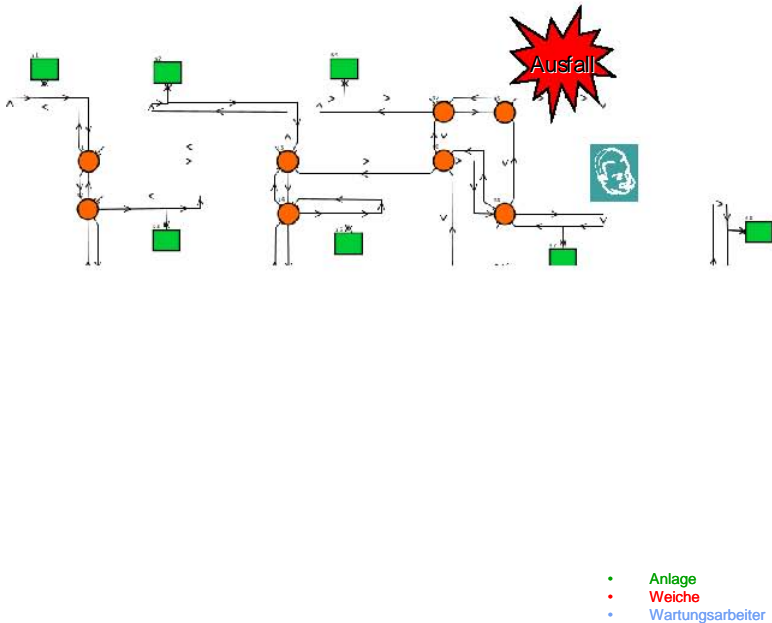


Abbildung 2-2: Layout einer Produktion inklusive Wartungsarbeitern

In dem Layout einer Produktion sind Anlagen (als grünes Quadrat dargestellt), deren Logistik mit Schienen und Weichen (als orangener Kreis dargestellt) und Wartungsarbeitern (als blaues Symbol dargestellt) abgebildet. Bei einem Anlagenausfall in der Produktion geht es darum, den Fehler möglichst schnell zu beheben und die Anlage sobald als möglich wieder in die Produktion einzugliedern. Wenn eine Anlage ausfällt, muss sie möglichst schnell wieder in einen produktiven Zustand versetzt werden. Um den Ausfall der Anlage den verfügbaren Wartungsarbeitern mitzuteilen, treten folgende Herausforderungen auf, die bei der Konzeption einer hochdynamischen Nachrichtenübermittlung erfüllt werden müssen:

- Welche Wartungsarbeiter sind in der Lage, das Problem zu beheben?
- Wie viele Wartungsarbeiter sollen die Nachricht über den Ausfall der Anlage zugestellt bekommen?
- Wie schnell müssen die Wartungsarbeiter auf den Ausfall einer Anlage reagieren?

- Auf welchem Endgerät (mit welchem Protokoll) ist der Wartungsarbeiter zur Zeit des Anlagen-Ausfalls erreichbar?
- Wird die Nachricht visuell und inhaltlich an das Endgerät angepasst?
- Auf welchem Endgerät bevorzugt der Wartungsarbeiter die Entgegennahme von Nachrichten? (z.B. 9:00 – 17:00 Uhr: Nachricht auf PDA; 17:00 – 09:00 Uhr: SMS auf Mobiltelefon)
- Was passiert, wenn der Sender und der Empfänger der Nachricht unterschiedliche oder nicht vereinbare Präferenzen bei der Nachrichtenzustellung haben?

Eine weitere Herausforderung besteht darin, dass am Markt erhältliche Nachrichtenorientierte Verteilungsplattformen einen Namensdienst besitzen, der typischerweise einen Namensraum mit Baumstruktur aufspannt [Ander07]. Das folgende Beispiel zeigt auf, warum Baumstrukturen als Namensraum unflexibel und einschränkend sind. Man nehme an, dass in einer Produktion in einem Zwei-Schicht-Betrieb produziert wird. Die Wartungsmitarbeiter besitzen ein mobiles Gerät, um Anlagendaten unterwegs schneller auslesen und bewerten zu können. Nun wird ein mobiles Gerät nicht nur von einem Wartungsarbeiter, sondern von zwei Wartungsarbeitern zusammen benutzt, je einem Wartungsarbeiter pro Schicht. Somit sind unter derselben physikalischen Adresse verschiedene Wartungsarbeiter zu unterschiedlichen Zeiten erreichbar. In einem bisher üblichen Namensraum mit einer Baumstruktur müsste deshalb die Realität verfälscht werden und das mobile Gerät zweimal im Namensbaum erscheinen, jeweils den Wartungsarbeitern der verschiedenen Schichten zugeordnet. Abbildung 2-3 spiegelt diesen Sachverhalt innerhalb eines Namensraums mit Baumstruktur wieder.

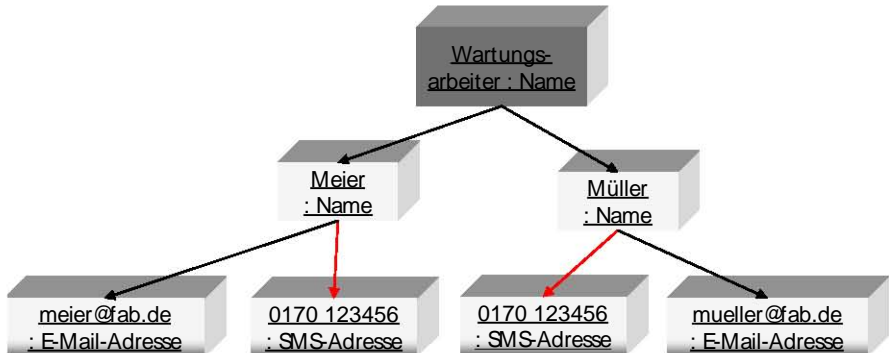


Abbildung 2-3: Namensraum als Baumstruktur

Die Wartungsarbeiter Meier und Müller haben jeweils eine E-Mail-Adresse und eine SMS-Adresse. Die SMS-Adresse beider Wartungsarbeiter besitzt dieselbe Telefonnummer und referenziert somit dasselbe Objekt. Nachteil bei der Baumstruktur ist, dass die Adresse des mobilen Gerätes zweimal eingetragen werden muss. Zusätzlich ist nicht direkt ersichtlich, dass es sich bei den Wartungsarbeitern Meier und Müller um dasselbe Gerät mit derselben Adresse handelt. Bei einem Defekt des mobilen Gerätes muss demnach auch das Gerät zweimal wieder aus dem Namensbaum ausgezogen werden. Im Gegensatz dazu ist in der Produktion gewünscht, dass die gemeinsame Adresse von Meier und Müller nur einmal abgebildet wird. Somit besteht die Anforderung nach einer

- **Abbildung von komplexen Beziehungen im Namensraum.**

Die nächste in dieser Arbeit verfolgte Herausforderung ist, dass je nach momentan verfügbaren Ein- und Ausgabegeräten und deren Eigenschaften bezüglich der Darstellung eine Nachricht möglichst optimal dem Empfänger zugestellt werden sollte. Wenn ein Empfänger gerade nur mit einem Mobiltelefon erreichbar ist, sollte eine Nachricht, wenn es der Inhalt zulässt, genau auf dieses Mobiltelefon zugestellt werden. Eine Nachrichtenorientierte Verteilungsplattform stellt Nachrichten nicht automatisch über einen **Protokollumwandler** (engl. Gateway) [Cham02] wie Email-Server, SMS-Server oder Voice-over-IP-Server zu, der die Nachrichten in das Protokoll des Empfangsgerätes konvertiert. Durch einen Protokollumwandler besteht die Möglich-

keit, die Nachricht über ein weiteres Netzwerk mit einem speziellen Protokoll zu versenden, welches für den Empfänger gerade optimal ist. Eine Automatisierung dieser Umwandlung würde zu einer hochdynamischen Nachrichtenübermittlung beitragen. Dies führt zu der Anforderung einer

- **automatischen Nachrichtenumwandlung in das Empfangsprotokoll.**

Eine wesentliche Herausforderung, um eine hochdynamische Nachrichtenübermittlung zu erreichen, ist dem Sender und Empfänger zu erlauben, Arbeitsabläufe bezüglich der Nachrichtenzustellung definieren und ausführen zu können. Um einen Arbeitsablauf ausführen zu können, muss die Nachricht an eine Arbeitsablaufverwaltungs-Komponente weitergeleitet werden. Dies führt zu einer zusätzlichen **Latenzzeit** bei der Nachrichtenübermittlung.

Wenn zwei Klienten nicht vereinbare Arbeitsabläufe ausführen lassen wollen, muss es eine **Überwachung** der Arbeitsabläufe geben. Das folgende Beispiel zeigt eine Notwendigkeit einer Abstimmung von Arbeitsabläufen.

Die Ausgangssituation ist ein Alarm einer Anlage in der Produktion. Auf diesen Alarm muss ein Wartungsarbeiter reagieren und persönlich den Status der Anlage vor Ort überprüfen. Es befinden sich die zwei Wartungsarbeiter Müller und Meier in der Produktionshalle, die den Alarm entgegennehmen können. Die Nachrichtenorientierte Verteilungsplattform sorgt dafür, dass bei einem Alarm die Gruppe Wartungsarbeiter eine Nachricht zugestellt bekommt. Jeweils der Sender und der Empfänger haben innerhalb der Nachrichtenorientierten Verteilungsplattform Arbeitsabläufe definiert. Der Arbeitsablauf des Empfängers sagt aus, dass nur ein Wartungsarbeiter den Alarm empfängt und zwar in der Reihenfolge zuerst Meier und dann als zweiter Müller. Somit ist laut diesem Arbeitsablauf Wartungsarbeiter Meier der eindeutige Empfänger der Nachricht. Der Sender des Alarms hat in einer Chronik die Information verfügbar, dass sich Wartungsarbeiter Müller erst vor einigen Minuten an der Anlage angemeldet hat. Somit wird der Alarm gesendet, mit dem Arbeitsablauf, dass Wartungsarbeiter Müller die Nachricht bekommen soll. Es treten somit die zwei folgenden widersprüchlichen Arbeitsabläufe auf:

- Arbeitsablauf des Empfängers: Nachrichtenzustellungsreihenfolge ist Meier vor Müller
- Arbeitsablauf des Senders: Nachrichtenzustellungsreihenfolge ist zuerst Müller

Um nicht vereinbare Arbeitsabläufe bei der Nachrichtenzustellung schnell und effizient aufzulösen, bedarf es einer Auflösung durch Regeln oder Prioritäten. Selbst wenn keine Regeln angegeben sind, sollte das System dennoch möglichst viele vereinbare Arbeitsabläufe ausführen können. Daraus lässt sich folgende Anforderung ableiten:

- **Schnelle Ausführung von Arbeitsabläufen von Sender und Empfänger.**

Zusammengefasst liegt der Fokus in dieser Arbeit darauf, ein Gesamtkonzept für Basiskomponenten einer Nachrichtenorientierten Verteilungsplattform zu konzipieren und zu implementieren, die den Anforderungen einer schnellen und hochdynamischen Kommunikation gleichzeitig gerecht werden und die aufgezeigten Herausforderungen bezüglich der Nachrichtenweiterleitung, der Nachrichtenzustellung und der Namensauflösung bei gleichzeitiger Ausführung von Arbeitsabläufen erfüllen.

Die Wichtigkeit von Nachrichtenorientierten Verteilungsplattformen wird von den Marktbeobachtungen der Softwarebranche untermauert. Laut Gartner Dataquest ist der **Markt** der Nachrichtenorientierten Verteilungsplattformen ein viel versprechender Markt innerhalb von Verteilungsplattformen und Enterprise Application Integration (EAI) [Tolle07, Meyer02, Ruh01] mit 160,7% Erlössteigerung innerhalb des Jahres 2005 [Gartner06]. Im Jahr 2005 wurden 8,5 Milliarden Dollar in diesem Bereich umgesetzt [Gartner06].

3 Stand der Technik

Es müssen Software- und Hardwarekomponenten gemeinsam **integriert** werden, damit ein standardisierter Nachrichtenaustausch zwischen Klienten wie z.B. Anlagen, Geräten, Applikationen, Systemen, Datenbanken, PCs und mobilen Geräten ermöglicht wird [Balzert01]. Die Systeme, Komponenten und Softwaretechniken, die dazu notwendig sind, werden in diesem Kapitel beschrieben.

3.1 Kommunikation

Das Prinzip der Kommunikation in der Informationstheorie ist auf dem **Kommunikationsmodell** Sender-Empfänger-Modell [Hall80, Hall97] begründet. In diesem Kommunikationsmodell wird eine Nachricht durch einen Übertragungskanal von einem Sender zu einem Empfänger übermittelt. Das **OSI-Schichtmodell** (Open Systems Interconnection Reference Model) [OSI94, ISO07] der Internationalen Organisation für Normung (ISO) beschreibt modellhaft die Kommunikation offener, informationsverarbeitender Systeme. Abbildung 3-1 zeigt die Kommunikation von einem Sender zu einem Empfänger im OSI-Schichtmodell.

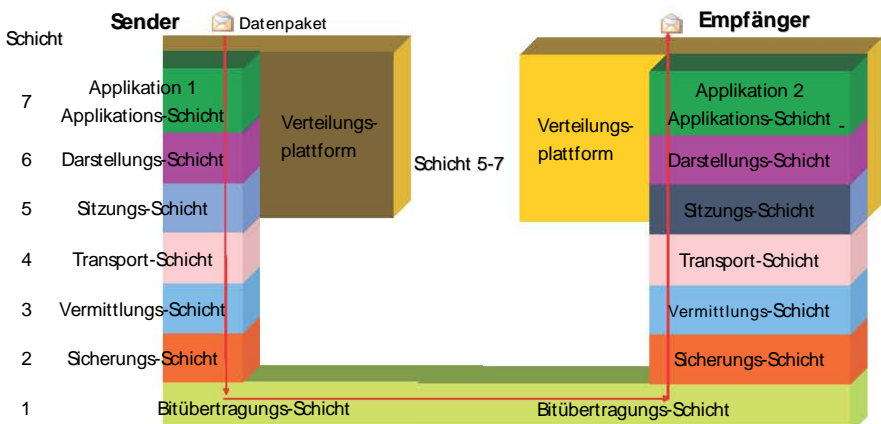


Abbildung 3-1: Kommunikation im OSI-Schichtmodell

Das OSI-Schichtmodell besteht aus sieben übereinander aufgebauten Schichten [Tan02, Weber98, Ha05]. Der Sender führt einen Funktionsaufruf der Applikation 1 in der **Applikations-Schicht** (Schicht 7) aus, um ein Datenpaket zu senden. Nun wird das Datenpaket beim Sender jeweils eine Schicht nach der anderen nach unten gereicht. Wenn der Empfänger das Datenpaket auf der **Bitübertragungsschicht** (Schicht 1) erhält, wird es jeweils eine Schicht nach der anderen nach oben weitergeleitet bis hin zur Applikations-Schicht, durch die der Empfänger auf den Nachrichteninhalte zugreifen kann. Jede Schicht im OSI-Schichtmodell legt Anforderungen fest, die in einer Instanz definiert sind. Die Instanzen auf jeder Schicht von Sender und Empfänger müssen denselben Regeln unterstellt sein, damit es eine Übereinstimmung in der Verarbeitung der Daten gibt. Eine Verbindung innerhalb der Instanzen einer Schicht wird somit aufgebaut und die Festlegung der Regeln innerhalb einer Schicht wird als **Protokoll** [Wiese03] definiert. Im Folgenden wird innerhalb der Kommunikation nur noch auf die Sitzungsschicht, die Darstellungsschicht und Applikations-Schicht eingegangen. Diese Schichten werden durch eine Verteilungsplattform [Ruh01, Tan95] verwirklicht.

Die Kommunikationsstrukturen in der Produktion bestehen aus verschiedenen Ebenen, die meist durch einen Protokollumsetzer (engl. Gateway) verbunden sind. Innerhalb der Ebenen wird normalerweise dasselbe Protokoll verwendet, zwischen den Ebenen kann sich das Protokoll unterscheiden [Lee06]. In Abbildung 3-2 ist der typische Aufbau einer Kommunikationsstruktur in der Produktion dargestellt.

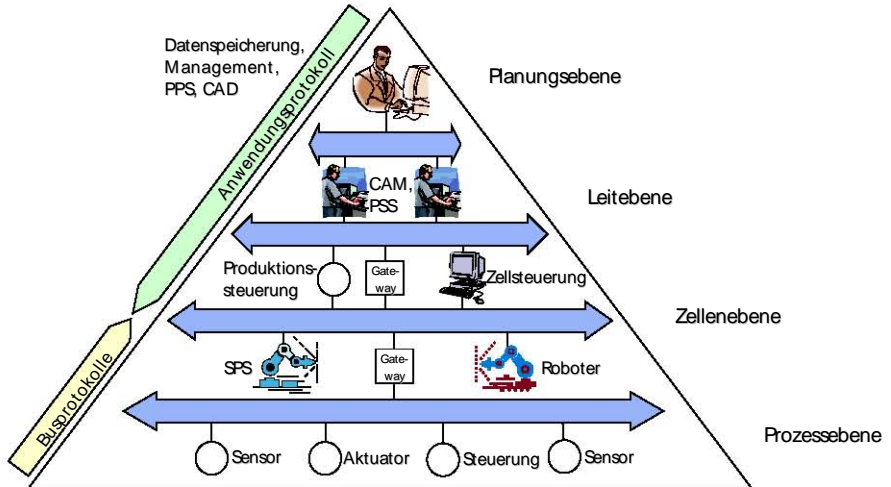


Abbildung 3-2: Kommunikationsstrukturen in der Produktion

Auf der Prozessebene werden Busprotokolle, wie z.B. das Feldbus-Protokoll eingesetzt, die einzelne Sensoren, Aktuatoren und Steuerungen verbindet. Auf der Zellebene sind Roboter, Anlagensteuerungen und Zellsteuerungs-Rechner durch entweder ein Busprotokoll oder ein Anwendungsprotokoll, wie z.B. Ethernet mit TCP/IP verbunden. Auf der Leitebene kommen PSS und CAM Anwendungen zum Einsatz, die genauso wie weitere Anwendungen auf der Planungsebene, wie z.B. Management-Anwendungen, Datenspeicherung, PPS oder CAD über ein Anwendungsprotokoll verbunden sind.

Bisher wurde die Kommunikation aus dem Schichtenmodell heraus beschrieben. Man benötigt mehrere Komponenten, die zusammengefügt die Kommunikation zwischen dem **Sender** und dem **Empfänger** ermöglichen. Im folgenden Beispiel aus der Produktion ist in Abbildung 3-3 der Sender eine Anlage und der Empfänger ein Produktionssteuerungssystem (PSS).

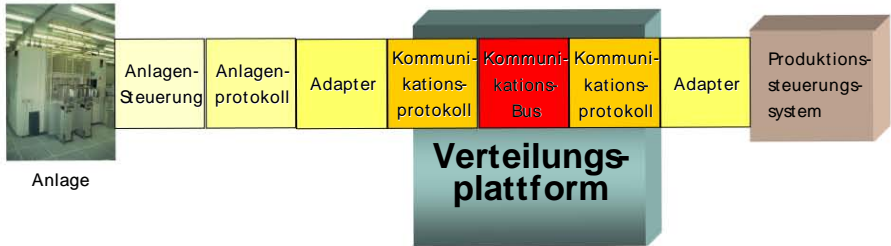


Abbildung 3-3: Komponenten der Kommunikation zwischen Sender und Empfänger

Die Anlage besitzt eine interne Steuerung, genannt Speicherprogrammierbare Steuerung (SPS) (engl. Programmable Logic Controller, Abk. PLC), die vom Anlagenhersteller programmiert wird. Über das Anlagenprotokoll werden die anlagenspezifischen Daten ausgetauscht. Beispiele für Anlagenprotokolle sind CAN, SECS/GEM oder Profibus. Durch einen Adapter wird das Anlagenprotokoll in das Kommunikationsprotokoll des Kommunikations-Busses übersetzt. Der Kommunikations-Bus besteht zum größten Teil aus der Verteilungsplattform, aber auch den darunter liegenden Schichten des OSI-Schichtmodells wie in Abbildung 3-1 dargestellt. Auf die Kommunikationsprotokolle wird in Kapitel 3.3 und 3.4 näher eingegangen.

Zusätzlich gibt es vier nichtfunktionale Anforderungen an die Kommunikation in der Produktion [Sieg92, Meyer02], die sich durch alle Schichten ziehen und beachtet werden müssen:

- Zuverlässigkeit
- Sicherheit
- Effizienz, Skalierbarkeit
- Deterministische Reihenfolge der Nachrichten

3.2 Verteilungsplattform

Eine **Verteilungsplattform** wird als Software beschrieben, die zwischen einer Anwendung und dem Netzwerk vermittelt [Howe07]. Die Verteilungsplattform bezeichnet in der Informatik anwendungsunabhängige Technologien, die Dienstleistungen zur Vermittlung zwischen Anwendungen anbieten, so dass die Komplexität der

zugrundeliegenden Infrastruktur, der Betriebssysteme, der Netzwerkprotokolle und der Hardware verborgen wird [Ruh01, Ulmer05]. Abbildung 3-4 zeigt die Abgrenzung einer Verteilungsplattform.

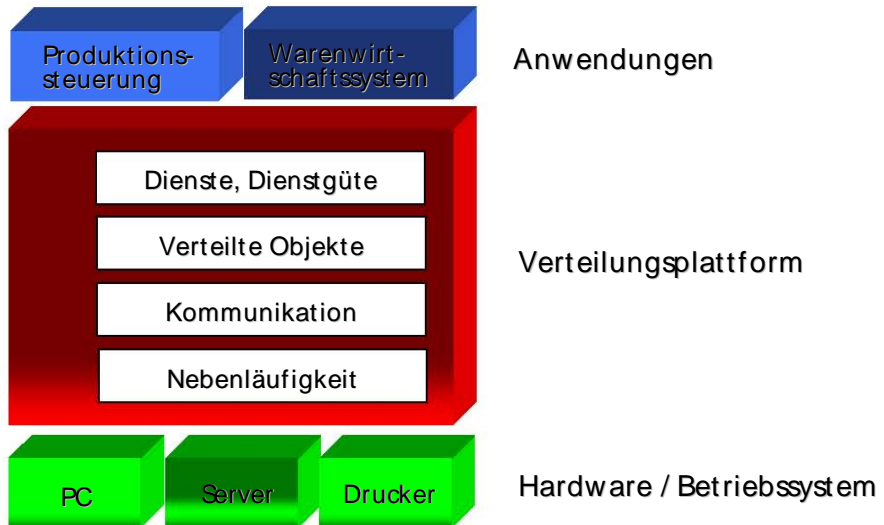


Abbildung 3-4: Abgrenzung einer Verteilungsplattform

Die Verteilungsplattform vermittelt zwischen beliebig vielen Anwendungen wie z.B. der Produktionssteuerung und dem Warenwirtschaftssystem und beliebig vielen Betriebssystemen oder Hardware Ressourcen wie z.B. PCs, Server und Drucker. Innerhalb der Verteilungsplattform befinden sich Dienste und Komponenten für die Dienstgüter, Verteilte Objekte, die Kommunikation und Nebenläufigkeit [Schantz03]. Eine Verteilungsplattform kann nach verschiedenen Prinzipien kategorisiert werden. Eine vollständige **Kategorisierung** von Verteilungsplattformen wurde von dem Middleware Resource Center¹ in Tabelle 1 definiert.

¹ <http://www.middleware.org/index.html> (02.07.2007)

Verteilungsplattform	Beschreibung
Nachrichtenorientierte Verteilungsplattform [Emm00]	Beruht auf der asynchronen Kommunikation, also der Übertragung von Nachrichten.
Objektorientierte Verteilungsplattform [Schmidt98, Heer99]	Besteht hauptsächlich aus Objekt Request Brokers (ORB) [CORBA04, Siegel98]. Beruht historisch auf der synchronen Kommunikation.
Prozedurale Verteilungsplattform [Srin95, Mul94]	Bietet Fernaufrufe auf ein entferntes System. Grundlage ist der Remote Procedure Call (RPC), der auf der synchronen Kommunikation beruht.
Datenbankenbasierte Verteilungsplattform [Grey93]	Ermöglicht direkten Zugriff auf die Datenstrukturen in einer Datenbank.
Transaktionsbasierte Verteilungsplattform [Weik02]	Besteht aus traditionellen Transaction Processing Monitors (TPM) [Gray93] und Web Application Servers [Alonso04].
Portal [Spath04, Vlach05, Gurz03]	Bietet eine Interaktion zwischen der Benutzeroberfläche und Systemen/Diensten.
Grid Verteilungsplattform [Foster04]	Infrastruktur, die eine integrierte, gemeinschaftliche Verwendung von meist geographisch auseinander liegenden, autonomen Ressourcen erlaubt.
Peer-to-Peer Verteilungsplattform [Choksi01, Gupta04]	Kommunikation von gleichberechtigten Systemen, hier bezogen auf ein Netzwerk von Computern.

Tabelle 1: Kategorien von Verteilungsplattformen

3.2.1 Marktanalyse über Verteilungsplattformen

Der Markt für Nachrichtenorientierte Verteilungsplattformen ist ein viel versprechender Markt innerhalb des Gebiets der Verteilungsplattformen und der Enterprise Application Integration. Innerhalb 2005 wurde eine **Erlössteigerung von 160,7%** erreicht, gegenüber einer Erlössteigerung von 7,1% des gesamten Marktes von Verteilungsplattformen und EAI [Gartner06]. Das weltweite Marktvolumen von Verteilungsplattformen und EAI war im Jahr 2004 7,9 Milliarden Dollar [Gartner05, Gartner06] und im Jahr 2005 8,5 Milliarden Dollar [Gartner06]. Im Jahr 2006 wurden mehr als **10 Milliarden Dollar** umgesetzt [Gartner07]. Es wird erwartet, dass der Markt jährlich um 12,9% bis auf 21 Milliarden Dollar im Jahr 2011 ansteigt [Gartner07]. Der Markt von Nachrichtenorientierten Verteilungsplattformen soll von 488,9 Millionen Dollar im Jahr 2004 auf 867,3 Millionen Dollar im Jahr 2011 anwachsen [Gartner02, Gartner05]. In Abbildung 3-5 ist die Marktentwicklung von Verteilungsplattformen und im speziellen Nachrichtenorientierten Verteilungsplattformen in Millionen Dollar dargestellt.

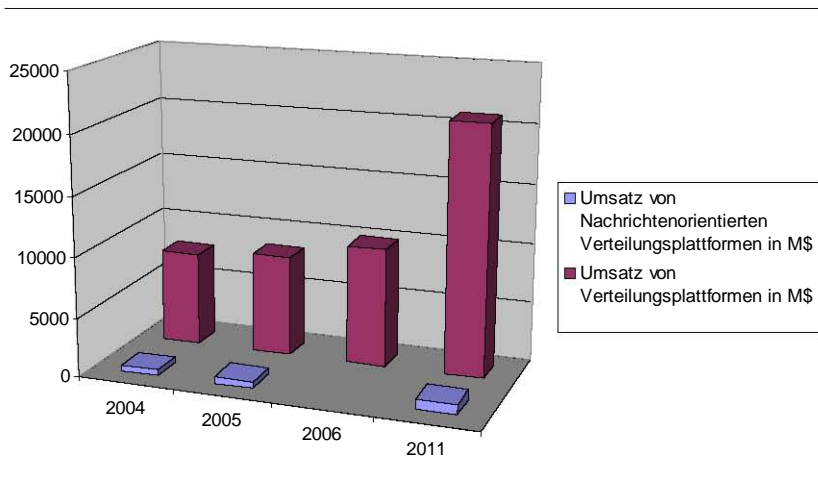


Abbildung 3-5: Marktentwicklung von Verteilungsplattformen

Der Markt für Verteilungsplattformen und EAI wurde 2005 von wenigen Anbietern bestimmt [Gartner06]. Der Marktführer IBM hatte 37,2% Erlösanteil, gegenüber den Verfolgern BEA Systems, Oracle, Microsoft und Tibco, die zusammen 31,5% Erlösanteil hatten. Alle weiteren Anbieter kommen zusammen nur auf 31,3% Erlösanteil. In Abbildung 3-6 sind die Marktanteile der erfolgreichsten Anbieter von Verteilungsplattformen dargestellt.

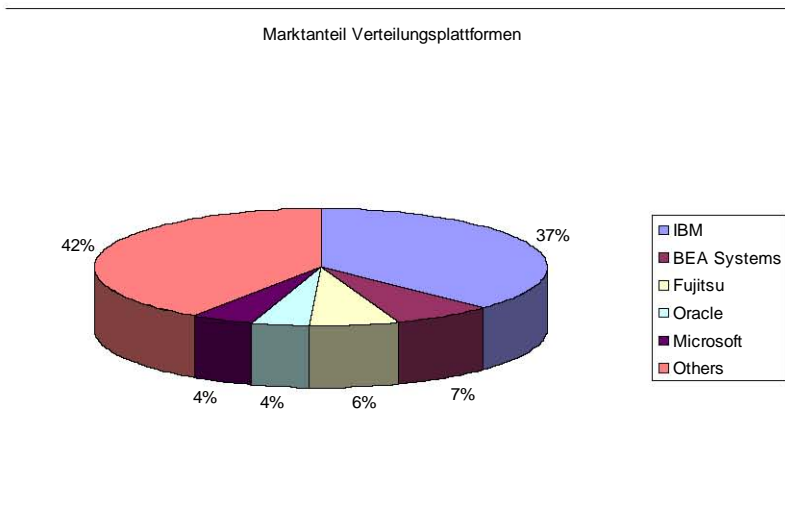


Abbildung 3-6: Marktanteil von Verteilungsplattformen

Innerhalb einer Untersuchung der Auswirkung des Einsatzes einer Nachrichtenorientierten Verteilungsplattform auf den Unternehmenserfolg in der Produktion wurden Geschäftsprozesse in der Halbleiterproduktion analysiert, um daraufhin Optimierungsmaßnahmen mit qualitativen und quantitativen Kriterien hinsichtlich des Unternehmenserfolges abzuleiten. Nach der Durchführung einer kompletten Auswirkungsanalyse der funktionalen Eigenschaften auf die Geschäftsprozesse ergab sich ein gesamtes Einsparpotential von 8,83% [Sezcan07].

3.2.2 Bewertung von Verteilungsplattformen

Die wichtigsten Funktionalitäten und Unterscheidungsmerkmale von Verteilungsplattformen in der Produktion sind die Art der Kommunikation, der Parallelisierbarkeit, der Art der Kopplung, der Verfügbarkeit, der Ausführungsgeschwindigkeit, der Veränderbarkeit, der Portabilität, der Flexibilität, der Geschwindigkeit und der Programmieraufwand [Tibco03]. Die Funktionalitäten von Verteilungsplattformen sind in Tabelle 2 zusammengefasst.

Verteilungs- plattform/ Funktionalität	Nachrichtenorientierte Verteilungsplattform	Objekt-orientierte Verteilungsplattform	RPC Verteilungs- plattform	Datenbankenbasierte Verteilungsplattform	Transaktionen Verteilungsplattform	Portal	Grid Verteilungsplatt- form	Peer-to-Peer Verteil- ungsplattform
Kommunikation	asyn- chron	syn- chron	syn- chron	syn- chron	syn- chron	asyn- chron	asyn- chron	asyn- chron
Parallele Nachrichten	+	O	-	+	+	+	+	+
Kopplung	lose	lose	eng	eng	eng	lose	eng	lose
Dienst muss verfügbar sein	+	-	-	-	-	+	-	+
Schnelle Ausführung	+	O	O	+	+	-	+	O
Toleranz für Veränderungen	+	O	-	-	-	+	O	+
Verfügbarkeit	+	-	-	-	-	+	O	+
Portabilität	+	+	-	-	-	+	O	O
Flexibilität	+	O	-	-	-	+	O	+
Geschwindig- keit	+	O	+	+	+	-	+	-
Programmier- Aufwand	-	-	+	O	O	O	-	O

Tabelle 2: Funktionalitäten von Verteilungsplattformen

Die Bewertung der Funktionalitäten der Verteilungsplattformen wurde vereinfacht auf folgende Darstellung:

- + Volle Erfüllung der Funktionalität, positive Bewertung
- O neutrale Bewertung
- Funktionalität wird nicht erfüllt, negative Bewertung

Als Verteilungsplattform für die Massenproduktion ist, aufgrund der hohen Anzahl von Klienten und Nachrichten, die Anforderung an einen hohen Nachrichtendurchsatz unverzichtbar. Die Entscheidung für die Art der passenden Verteilungsplattform kann nicht verallgemeinert werden, sondern hängt von der Art des Produktionsprozesses ab. In dieser Arbeit handelt es sich um eine Hightech Massenproduktion, wie z.B. der Halbleiterproduktion welche komplexe Produktionsprozesse mit sich bringt. Da ein einzelnes Produkt bis zu 90 Tage in der Linie einer Fertigung zubringt und bis zu 500 Prozessschritte auf über 120 verschiedenen Anlagen möglich sind, stoßen verschiedene bestehende Verteilungsplattformen beim Nachrichtendurchsatz an ihre Systemgrenzen [Mönch03]. Abbildung 3-7 zeigt einen typischen Halbleiter-Produktionsprozess in der Massenproduktion, der als Grundlage für Produktionsprozesse in dieser Arbeit verwendet wird.

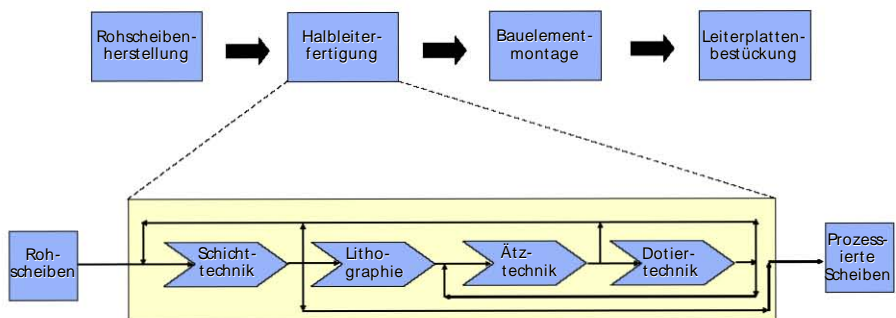


Abbildung 3-7: Prozessschritte in der Halbleiterfertigung

Von der Rohscheibenerstellung bis zum verkaufsfertigen Halbleiterelement werden Produktionsprozesse innerhalb von vier verschiedenen Fertigungen durchlaufen. Dabei besteht die Halbleiterfertigung aus der Schichttechnik, der Lithographie, der Ätztechnik

und der Dotiertechnik, die in Schleifen durchlaufen werden. Bei solchen hochkomplexen Produktionsprozessen, ist es sinnvoll diesen zu entkoppeln und damit eine **asynchrone Kommunikation** [Tan95] zu verwenden. Somit wird in dieser Arbeit eine **Nachrichtenorientierte Verteilungsplattform** einer Objektorientierten Verteilungsplattform oder einer Prozeduralen Verteilungsplattform vorgezogen. Bestehende Applikationen wie Produktionssteuerungs- oder Produktionsplanungssysteme, die mit einer Verteilungsplattform verbunden sind, bestehen nicht immer aus einer Datenbank. Somit sollte eine Datenbankenbasierte Verteilungsplattform oder Transaktionsbasierte Verteilungsplattform nur verwendet werden, wenn alle Systeme auf der Basis von Datenbanken arbeiten. In dieser Arbeit wird auf ein möglichst breites Anwendungsspektrum innerhalb der Massenproduktion geachtet und dadurch sind die Datenbankenbasierten Verteilungsplattformen und Transaktionsbasierten Verteilungsplattformen weniger geeignet als eine Nachrichtenorientierte Verteilungsplattform. Da alle Verteilungsplattformen in der Produktion hauptsächlich Nachrichten von Anlagen weiterleiten, gibt es kaum eine Benutzerinteraktion, für die ein Portal ausgelegt ist. Somit ist ein Portal für die Produktion weniger geeignet als eine Nachrichtenorientierte Verteilungsplattform. Eine Grid Verteilungsplattform bietet zu wenig Transparenz und keine Standards und wird deshalb nicht einer Nachrichtenorientierten Verteilungsplattform vorgezogen. Bei einer Peer-to-Peer Verteilungsplattform kann keine Verfügbarkeit oder Verbindungsqualität der Peers vorausgesetzt werden, weswegen diese Art der Verteilungsplattform für die Produktion nicht in Frage kommt. Daher wird als Verteilungsplattform für die Konzeption und Entwicklung eine Nachrichtenorientierte Verteilungsplattform ausgewählt.

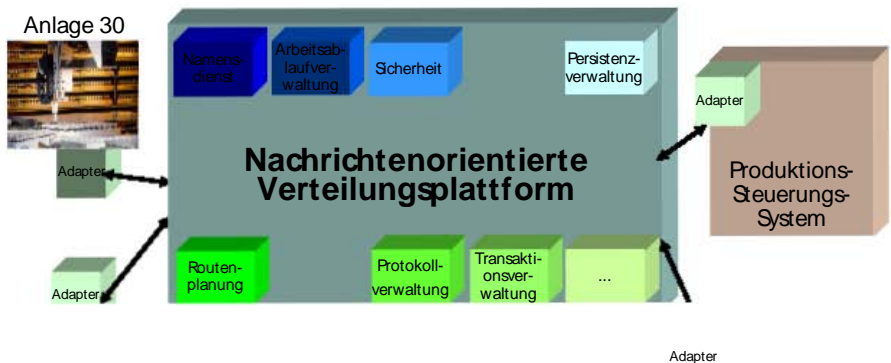
Typische, am Markt erhältliche Verteilungsplattformen, die in der Produktion eingesetzt werden, sind in Anhang B aufgelistet.

3.3 Nachrichtenorientierte Verteilungsplattform

Eine Nachrichtenorientierte Verteilungsplattform bezeichnet eine Verteilungsplattform, die auf der asynchronen Kommunikation, also der Übertragung von Nachrichten beruht [Emm00]. Das Format für die Nachrichten ist nicht festgelegt, in der Praxis hat sich jedoch XML als ein beliebtes Format etabliert und wird in der Produktion immer öfter

eingesetzt [Dreiss05, Meier06]. Kommunikationsprotokolle der Nachrichtenorientierten Verteilungsplattform sind der direkte Nachrichtenaustausch, die Punkt-zu-Punkt Nachrichtenübermittlung über Warteschlangen und das Veröffentlichen und Abonnieren [Emm00, Ha05].

Als notwendige **Komponenten** einer Nachrichtenorientierten Verteilungsplattform gelten der **Namensdienst** [Tan02] und die **Routenplanung** [O'Reil04, Tibco03, Knoll05]. Typischerweise beinhalteten die am Markt verfügbaren Nachrichtenorientierten Verteilungsplattformen die Komponenten Namensdienst, Routenplanung, **Arbeitsablaufverwaltung** [Aalst02, Spath06a], **Sicherheit** [Ford05], **Persistenzverwaltung** [Harrop05, Walls05], **Transaktionsverwaltung** [Mahm04] und **Protokollverwaltung** [Wiese03]. Folgende Abbildung 3-8 spiegelt die typischen Komponenten einer Nachrichtenorientierten Verteilungsplattform wieder.



Anlage 29

Anlage 30

Abbildung 3-8: Nachrichtenorientierte Verteilungsplattform

Damit sich eine Anlage oder Applikation mit der Nachrichtenorientierten Verteilungsplattform verbinden kann, benötigt man einen **Adapter** [Knoll05], um von dem Protokoll der Anlage oder der Applikation auf das Protokoll der Nachrichtenorientierten Verteilungsplattform zu übersetzen. Durch diese Verbindung wird eine Nachrichtenübermittlung erst ermöglicht.

Die Komponente **Namensdienst** verwaltet alle physikalischen Adressen, die mit der Nachrichtenorientierten Verteilungsplattform verbunden sind und löst diese als Namen auf. Durch die Objekte des Namensdiensts wird ein Namensraum mit hierarchischer Ordnung aufgespannt [Weber98].

Die **Routenplanung** dient zur Weiterleitung von Nachrichten innerhalb der Nachrichtenorientierten Verteilungsplattform [O'Reil04, Tibco03].

Um Arbeitsabläufe ausführen zu können, benötigt man eine geeignete Ausführungsumgebung [Aalst02, Spath06a]. Die **Arbeitsablaufverwaltung** stellt solch eine Ausführungsumgebung dar.

Die **Sicherheitskomponente** bietet die grundlegenden Sicherheitsdienste zur Authentisierung, Autorisierung und Verschlüsselung an. Bestehende Sicherheitsbibliotheken wie Java Authentication and Authorization Service (JAAS) [JAAS01], Java Generic Security Services (Java GSS-API) [GSSAPI02] oder Java Cryptography Extension (JCE) [JCE04] sind frei verfügbar und decken Sicherheitsrisiken in der Produktion ab [Berger05].

Die Komponente **Persistenzverwaltung** ist für die langfristige Speicherung der Daten in einer Datenbank zuständig [Tan02, Degen01]. Der Anwendung wird eine Schnittstelle zur Datenbank angeboten, in der die transienten Daten des Hauptspeichers in ein geeignetes Speicherformat umgewandelt und persistent gespeichert werden können. Der Namensraum des Namensdienstes muss vor allem persistent gespeichert werden, damit die physikalischen Adressen von Anlagen, Geräten und Applikationen nicht immer von neuem eingetragen werden müssen. Ebenso werden Regeln für Arbeitsabläufe persistent gespeichert, um diese wiederverwenden zu können. Wenn eine Nachrichtenorientierte Verteilungsplattform als Verteiltes System [Tan02] arbeitet, ist eine verteilte Persistenz [Degen01] erforderlich, die durch Technologien wie Enterprise Java Beans (EJB) [Ihns03], Java Data Objects (JDO) [Russell06], Hibernate [Hiber07] oder ADO.NET [Mauerer05] zur Verfügung gestellt werden.

Die **Transaktionsverwaltung** wird meist zusammen mit der Persistenzverwaltung verwendet, um Zugriffe auf eine Datenbank in Transaktionen [Mahm04] zu unterteilen und damit gegen Systemausfälle zu sichern. Innerhalb einer Nachrichtenorientierten

Verteilungsplattform greifen Anwender typischerweise parallel auf Daten zu und verändern diese. Um die Datenkonsistenz bei diesen Zugriffen zu bewahren, werden Transaktionen eingesetzt, die den sogenannten ACID Eigenschaften [Mahm04] entsprechen. Diese ACID Eigenschaften sind definiert durch Atomarität, Konsistenz, Isolation und Dauerhaftigkeit. Durch die Transaktionsverwaltung kann das System den Ausfall eines Teilsystems kompensieren und den Zustand vor dem Ausfall wiederherstellen.

Die **Protokollverwaltung** stellt der Nachrichtenorientierten Verteilungsplattform eine Anzahl an verschiedenen Protokollen zur Umwandlung einer Nachricht zur Verfügung. Um Daten mit den Klienten auszutauschen, die mit der Nachrichtenorientierten Verteilungsplattform verbunden sind, müssen die Datentypen des Klientenprotokolls an die Datentypen der Nachrichtenorientierten Verteilungsplattform angepasst werden [Knoll05].

In der Produktion werden typischerweise die Protokolle:

- SOAP [SOAP07],
- HTTP [HTTP99],
- SECS/GEM [SemiE37-06],
- Object RPC (DCOM-Protokoll) [Heer99],
- OPC [OPC06],
- Profibus-Protokoll [Popp00],
- CAN-Protokoll,
- IIOP [CORBA04],
- TCP/IP [TCP89] und
- RS232-Protokoll [Cam88]

eingesetzt. Es gibt folgende Standards JMS [JMS07], JCA [JCA07], COM Interface², .NET Interface (ASP.NET)³ für Programmier-Schnittstellen (engl. Application Programming Interface, Abk. API), die eine Basis darstellen, um Protokolle zu definieren.

² <http://www.microsoft.com/com/> (17.07.2007)

³ <http://www.microsoft.com/net/> (17.07.2007)

Innerhalb der Produktion werden zur Kommunikation zwischen Sensoren und Aktuatoren verschiedenste Sensor Protokolle eingesetzt [Choksi01, Heide01, Heinzel00, Intana00].

Die Automobilindustrie verwendet hauptsächlich Feldbusprotokolle [Lawrenz00]. Beispiele für Feldbusse in der Produktion sind Profibus [Popp00], Interbus-S⁴, P-Net⁵, Local Operating Network (LON)⁶, Factory Implementation Protocol (FIP)⁷, Aktuator Sensor Interface (ASI)⁸ und Bitbus⁹.

Obwohl Feldbusse in der Fertigung Echtzeitanforderungen erfüllen und seit den späten 90ern als internationaler Standard anerkannt wurden, leiden sie unter hohen Hardware- und Softwarekosten und ungewisse Kompatibilität mit anderen Systemen [Lee06]. Diese Defizite verhindern einen Einsatz von Feldbussen in verschiedenen Fertigungen [Kaplan01]. Als Alternative wird Ethernet wegen seiner Einfachheit und Akzeptanz immer häufiger eingesetzt [Park98]. Durch die Benutzung von Vollduplex können Datenzusammenstöße eliminiert werden und machen somit Echtzeitfähige Industriernetzwerke auf Ethernet-Basis möglich [Stall00].

In der Halbleiterfertigung wird momentan noch die Proprietäre Schnittstelle SECS/GEM eingesetzt [Foy04]. Da dieses Protokoll keinen Standard für die Datenrepräsentation besitzt, wird die Anlagenkommunikation auf Interface A, ein XML-basiertes Protokoll umgestellt.

3.3.1 Verteilte Nachrichtenorientierte Verteilungsplattform

Für den Aufbau einer Nachrichtenorientierten Verteilungsplattform muss die passende Softwarearchitektur gewählt werden, um eine schnelle und hochdynamische Nachrichtenübermittlung zu erreichen. Bei der Auswahl der passenden Softwarearchitektur für eine Nachrichtenorientierte Verteilungsplattform muss die Zentralistische Softwarear-

⁴ <http://www.interbusclub.com/> (17.07.2007)

⁵ <http://www.p-net.org/> (17.07.2007)

⁶ <http://www.lonmark.org/> (17.07.2007)

⁷ <http://www.worldfip.org/> (17.07.2007)

⁸ <http://www.as-interface.net/> (17.07.2007)

⁹ <http://www.bitbus.org/> (17.07.2007)

chitektur mit der Verteilten Softwarearchitektur verglichen werden [Tan02]. Ein Verteilte Softwarearchitektur zeichnet sich vor allem dadurch aus, dass einzelne Komponenten räumlich verteilt und über ein Kommunikationsnetzwerk miteinander verbunden sind [Weber98, Ha05, Tan02]. Ein **Verteiltes System** wird definiert als Zusammenschluss unabhängiger Computer, die sich für den Benutzer als ein einzelnes System präsentiert [Tan02]. Die Verfügbarkeit des Systems wird dadurch erhöht, dass eine nahezu beliebige Anzahl an Hardwarekomponenten und Servern verwendet werden kann. Somit kann zum Beispiel bei Überschreitung der Kapazitätsgrenze von CPU, RAM oder Netzwerkbandbreite eines Servers, ein neuer Server im System die Last übernehmen. Allerdings gilt, je höher die Anzahl an Servern im System ist, umso mehr steigt der Synchronisationsaufwand zwischen den Servern und dadurch kann die Gesamtgeschwindigkeit des Systems sinken.

Eine gebräuchliche Definition von Verteilungsplattformen ist, dass sie auf einem Verteilten System aufsetzt und eine Programmierschicht zwischen Anwendung und dem zugrundeliegenden Betriebssystem, den Netzwerkprotokollen und der Hardware darstellt [Ha05]. In der Praxis wird davon ausgegangen, dass eine Nachrichtenorientierte Verteilungsplattform über einem Verteilten System installiert wird. In Abbildung 3-9 sind beispielhaft vier Nachrichtenorientierte Verteilungsplattformen dargestellt, die ein Verteiltes System bilden und sich untereinander **synchronisieren**.

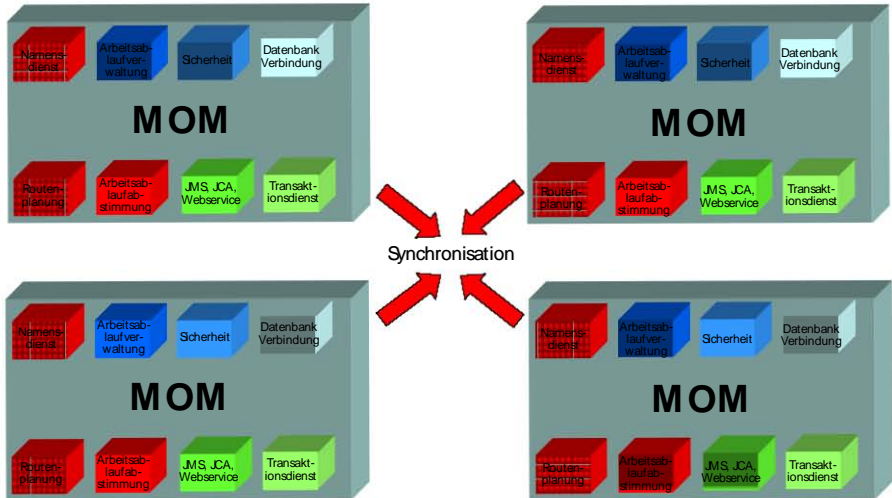


Abbildung 3-9: Synchronisation der Nachrichtenorientierten Verteilungsplattform im Verteilten System

Die vier Nachrichtenorientierten Verteilungsplattformen erreichen durch die Synchronisation, dass alle Komponenten untereinander auf den aktuellen Stand gebracht werden. Auf die einzelnen Komponenten wurde in Kapitel 3.2 eingegangen. Eine verteilte Nachrichtenorientierte Verteilungsplattform steht dem Anwender gegenüber **transparent** als ein Gesamtsystem zur Verfügung [Leser07]. Die Koordination der Replikation von dynamisch adaptiven Systemen ist in einem Verteilten System eine Herausforderung, hilft aber die Performanz zu erhöhen [Jacobsen04].

3.3.2 Bewertung von Nachrichtenorientierten Verteilungsplattformen

Zusammenfassend werden die **Vorteile** einer Nachrichtenorientierten Verteilungsplattform dargestellt [Mahm04, Ha05]:

- Die Möglichkeit einer asynchronen Kommunikation.
- Ein Dienst für die Nachrichtenübermittlung muss nicht sofort verfügbar sein.
- Es sind Warteschlangen für Nachrichten (engl. Message-Queues) vorhanden.

- Eine meist schnellere Ausführung als in Funktionsaufruf-basierten Programmen.
- Eine lose Kopplung von Server und Klienten besteht.
- Mehr Toleranz für Änderungen der bestehenden Funktionen.
- Eine verbesserte Verfügbarkeit der Systeme.
- Eine parallele Verarbeitung von Nachrichten ist möglich.
- Eine Erhöhung der Kompatibilität.
- Eine Erhöhung der Portabilität.
- Eine Erhöhung der Flexibilität einer Applikation.
- Eine Reduzierung der Komplexität einer Applikation.
- Die Möglichkeit der Integration verschiedener Systeme.
- Eine Nachrichten-Zustellungsgarantie ist vorhanden.
- Eine flexible Adressierung ist möglich.
- Eine Erhöhung des Nachrichtendurchsatzes im System.

Demgegenüber gibt es folgende **Nachteile** einer Nachrichtenorientierten Verteilungsplattform [Mahm04, Ha05]:

- Eine hohe Komplexität beim Programmieren.
- Ein hoher Aufwand bei der Erstellung von Applikationen.
- Ein Ausfall legt alle angeschlossenen Systeme lahm.
- Das Design, das Testen, das Debuggen und die Entwicklung der Komponenten sind für Synchron-Programmierer ungewohnt.

Die Bewertung, ob eine **Zentralistische Softwarearchitektur oder Verteilte Softwarearchitektur** in der Produktion eingesetzt werden soll, hängt von der Anzahl der Klienten und Nachrichten pro Sekunde im System ab. Bei vielen Nachrichtenorientierten Verteilungsplattformen wird empfohlen, für jeweils 20 Klienten einen Server zur Verfügung zu stellen [JBoss07].

- Verteilte Systeme haben durch billige Mikroprozessoren ein besseres Preis-/Leistungsverhältnis als ein Großrechner.

- Ein Verteiltes System kann mehr Rechenleistung bereitstellen als ein Großrechner.
- Rechenleistung kann in einem Verteilten System durch kleine Erweiterungen hinzugefügt werden.
- Wenn ein Computer abstürzt, kann ein Verteiltes System dennoch überleben.
- Rechenlast kann kosteneffektiv über alle verfügbaren Computer verteilt werden.
- Der Ausfall des Netzwerkes kann in einem Verteilten System größere Auswirkungen haben als bei einem Großrechner.
- Der Programmier-Aufwand ist bei einem Verteilten System höher als bei einem Zentralistischen System.

3.4 Kommunikationsmodelle einer Nachrichtenorientierten Verteilungsplattform

Innerhalb einer Nachrichtenorientierten Verteilungsplattform gibt es drei unterschiedliche Modelle der Kommunikation. Diese sind der direkte Nachrichtenaustausch, die Punkt-zu-Punkt Nachrichtenübermittlung über Warteschlangen und das Veröffentlichen und Abonnieren (engl. Publish and Subscribe) [Tanner04].

Der **direkte Nachrichtenaustausch** ist eine Kommunikationsform, die auf dem Versenden von Nachrichten von einem Sender zu einem Empfänger beruht [Schrö03, Krogh95]. Sie wird innerhalb der Nebenläufigkeits-Programmierung, der parallelen Programmierung, der objektorientierten Programmierung und der Interprozesskommunikation eingesetzt. Beim direkten Nachrichtenaustausch muss für jede Verbindung zwischen einem Sender und einem Empfänger jeweils eine Übersetzung zwischen zwei Schnittstellen implementiert werden. In Abbildung 3-10 ist der direkte Nachrichtenaustausch zwischen zwei Klienten dargestellt.



Abbildung 3-10: Direkter Nachrichtenaustausch

Ein Klient sendet die Nachricht zu einem zweiten Klient, der den Erhalt der Nachricht bestätigt. Für jeden Klienten, der der Nachrichtenorientierten Verteilungsplattform hinzugefügt wird, muss eine Übersetzung auf jeden anderen Klienten innerhalb der Nachrichtenorientierten Verteilungsplattform implementiert werden. Abbildung 3-11 zeigt sechs verschiedene Klienten mit ihren Verbindungen innerhalb einer Nachrichtenorientierten Verteilungsplattform.

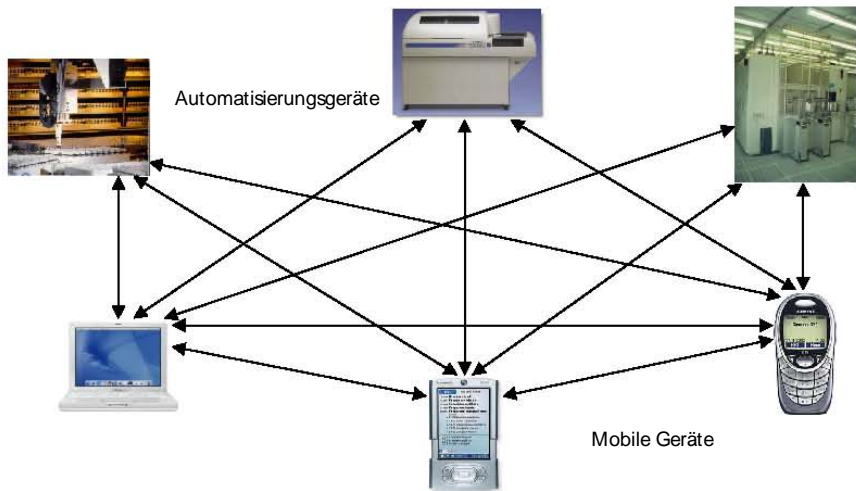


Abbildung 3-11: Direkter Nachrichtenaustausch mit mehreren Klienten

Wenn zwei Klienten miteinander kommunizieren wollen, so muss bereits zuvor eine Verbindung zwischen den Schnittstellen dieser Klienten implementiert worden sein.

Die **Punkt-zu-Punkt Nachrichtenübermittlung über Warteschlangen** bezeichnet die Nachrichtenübermittlung in Form von Warteschlangen [Tan95] und bietet Plattformunabhängigkeit, Skalierbarkeit und Persistenz. In **Fehler! Verweisquelle konnte nicht gefunden werden.** ist der Mechanismus der Punkt-zu-Punkt Nachrichtenübermittlung über Warteschlangen abgebildet.



Abbildung 3-12: Punkt-zu-Punkt Nachrichtenübermittlung über Warteschlangen

Wenn ein Klient mit einem anderen Klient kommunizieren will, sendet er eine Nachricht an eine Warteschlange der Nachrichtenorientierten Verteilungsplattform. Die Warteschlange bestätigt, dass die Nachricht an der Nachrichtenorientierten Verteilungsplattform angekommen ist. Sobald die Nachricht in der Warteschlange angekommen ist, wird sie dem Empfangs-Klienten zugestellt. Dieser empfängt die Nachricht und bestätigt der Nachrichtenorientierten Verteilungsplattform den Empfang. Typischerweise wird eine Punkt-zu-Punkt Verbindung physikalisch auf den OSI-Schichten 1-3 aufgebaut, wobei der Hauptanteil innerhalb der Vermittlungs-Schicht liegt. Das hier dargestellte Kommunikationsmodell liegt über der Transport-Schicht und wird teilweise Ende-zu-Ende Verbindung genannt.

Veröffentlichen und Abonnieren ist ein Entwurfsmuster aus dem Bereich der Softwareentwicklung. Es dient zur Weitergabe von Änderungen an einem Objekt an von diesem Objekt abhängigen Strukturen. Das Kommunikationsmodell Veröffentlichen und Abonnieren ist als 1:n Kommunikation für Gruppenruf Nachrichten (engl. Multicast) [William00, Hinden06] ausgelegt. In Abbildung 3-13 ist das Kommunikationsmodell Veröffentlichen und Abonnieren dargestellt.

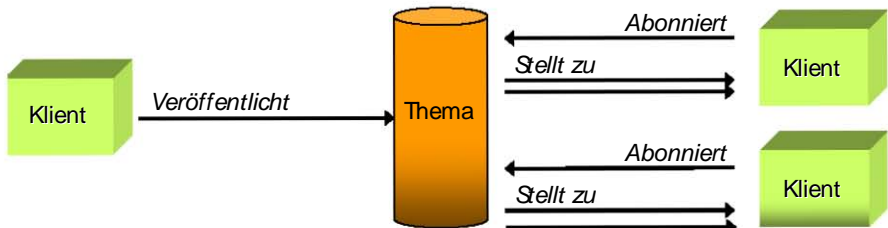


Abbildung 3-13: Veröffentlichen und Abonnieren

Um eine Kommunikation zwischen einem Sender und mehreren Empfängern herzustellen, müssen die Klienten, die Nachrichten von einem Sender empfangen wollen, sich zuerst das Thema des Senders abonnieren. Der Sender sendet seine

Nachricht an ein Thema. Das Thema stellt eine besondere Art der Warteschlange dar. Sobald die Nachricht beim Thema eintrifft, bekommt jeder Klient, der das Thema abonniert hat, die Nachricht zugestellt. Solange der Klient mit dem Thema verbunden bleibt, werden ihm alle Nachrichten zu diesem Thema zugestellt. Dies wird in der Abbildung durch den doppelten Pfeil vom Thema zum Klient dargestellt. Erst wenn der Klient die Abonnerung des Themas rückgängig macht, werden keine Nachrichten über das Thema mehr empfangen. Der Gruppenruf ist ebenfalls wie die Punkt-zu-Punkt Verbindung hauptsächlich auf den Schichten 1-3 des OSI-Schichtmodells angesiedelt. Das hier dargestellte Kommunikationsmodell liegt über der Transport-Schicht. Bei Veröffentlichen und Abonnieren gibt es subjektbezogene Systeme (engl. subject-based) [Wack97, Castro02, Zhuang01] und inhaltsbezogene Systeme (engl. content-based) [Wang04, Gupta04, Cheung06]. Die inhaltsbezogenen Systeme werden weiter unterschieden in abbildungsbasierte (engl. map-based) Systeme [Aguilera99] und XML-basierte Systeme [Pereira01].

3.4.1 Bewertung der Kommunikationsmodelle

Wenn eine Nachrichtenorientierte Verteilungsplattform innerhalb einer Produktion zum Einsatz kommt, muss das richtige **Kommunikationsmodell** gewählt werden. Bei der direkten Nachrichtenübermittlung ist der Aufwand zur Erstellung von den Schnittstellen zwischen den Klienten sehr hoch. Es müssen für jeden zum System hinzukommenden Klienten, Schnittstellen zu jedem anderen Klienten implementiert werden. Somit steigt die Anzahl der Schnittstellen exponentiell an. Dies ist aus Abbildung 3-11 ersichtlich, da bei n bestehenden Klienten mit $n \in \mathbb{N}$ ein hinzukommender $n+1$. Klient n neue Verbindungen aufbauen muss. Wenn man von mehreren hundert Anlagen und zusätzlich mehreren hundert Klienten in der Massenproduktion ausgeht, ist dieses Kommunikationsmodell ungeeignet.

Die Punkt-zu-Punkt Nachrichtenübermittlung über Warteschlangen hat den Vorteil, dass nur bei diesem Kommunikationsmodell der Benutzer die volle Kontrolle über die Nachricht und die Daten hat. Bei diesem Kommunikationsmodell ist es möglich, in die

Nachrichtenzustellung einzugreifen und Arbeitsabläufe¹⁰ [Aalst02] für Sender und Empfänger auszuführen. Nachteil bei der Punkt-zu-Punkt Nachrichtenübermittlung über Warteschlangen ist, dass die bestehenden Systeme bei der Geschwindigkeit der Nachrichtenübermittlung begrenzt sind [Stiedl04, Knoll06a]. Somit muss die Flexibilität der Punkt-zu-Punkt Nachrichtenübermittlung durch eine langsamere Nachrichtenübermittlung erkauft werden.

Das Kommunikationsprotokoll Veröffentlichen und Abonnieren wird oftmals in komplexen Produktionen eingesetzt, in denen eine schnelle Nachrichtenübermittlung unabkömmlich ist [M2A03b]. Dieses Kommunikationsprotokoll bietet zum einen eine Gruppenruf-Verbindung von einem Sender zu mehreren Empfängern, wodurch die Nachrichtenübermittlung beschleunigt wird. Zum anderen wird die Nachricht direkt vom Thema zum Empfänger weitergeleitet, ohne dass sie modifiziert werden muss. Dadurch ist die Nachrichtenübermittlung konstant schnell, büßt aber Flexibilität ein. Bei diesem Kommunikationsprotokoll gibt es keine Möglichkeit einer situationspezifischen Abstimmung der Klienten oder einer Zuordnungsverhandlung der Nachricht. Da die Anforderung besteht, dass Arbeitsabläufe von Sender und Empfänger ausgeführt werden sollen, wird als Kommunikationsprotokoll eine Punkt-zu-Punkt Nachrichtenübermittlung für die Entwicklung übernommen. Die daraus abgeleitete Anforderung lautet somit:

- **Punkt-zu-Punkt Nachrichtenübermittlung mit höherem Durchsatz als bei bestehenden Systemen.**

3.5 Namensdienst

Der **Namensdienst** ist eine notwendige Komponente einer Nachrichtenorientierten Verteilungsplattform [Tibco03]. Innerhalb einer Nachrichtenorientierten Verteilungsplattform stehen dem Anwender verschiedene **Ressourcen** zur Verfügung, die durch den Namensdienst identifiziert und lokalisiert werden. Bei diesen Ressourcen handelt es sich um Anlagen, Geräte, Computer, Dienste, Ports, Netzadressen, Objekte, Dateien,

¹⁰ <http://www.software-kompetenz.de/?2368&highlight=workflow>

Prozesse, Benutzer, Datenbanken oder Webservices. Im Namensdienst werden die Namen der Ressourcen veröffentlicht und ihnen eine Referenz zur **physikalischen Adresse** mit den benötigten Informationen zum Verbindungsaufbau zugeordnet. Einen Namen nachzuschauen und die Referenz zu erhalten bezeichnet man als **Namensauflösung** [Weber98]. Durch den Namensdienst wird ein **Namensraum** aufgespannt, der die Zuordnung von Namen zu physikalischen Adressen strukturiert [Tan02, Weber98]. Die Namen werden in einem Namensraum durch Objekte repräsentiert.

Ein Namensraum umfasst die Menge aller gültigen Namen entsprechend einer definierten Syntax. Typische Anwendungen eines Namensraums sind die hierarchisch angelegten Bezeichnungen von Dateiverzeichnissen oder Netzdomänen. Einen **hierarchischen Namensraum** erreicht man durch strukturierte Namen. Abbildung 3-14 zeigt einen hierarchischen Namensraum.

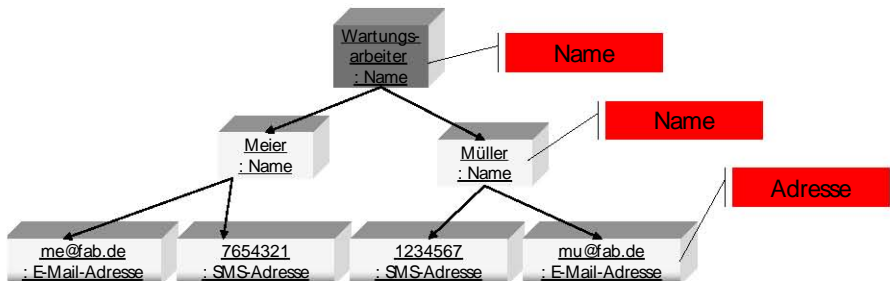


Abbildung 3-14: Hierarchischer Namensraum

Alle Knoten in dem Namensraum werden als Namen dargestellt, wie **Wartungsarbeiter Meier** und **Wartungsarbeiter Müller**. Alle Blätter beinhalten die Referenz zu der physikalischen Adresse mit den benötigten Informationen zum Verbindungsaufbau. Normalerweise werden Namensräume aufgrund der hierarchischen Ordnung durch eine Baumstruktur dargestellt.

Ein **Baum** ist in der Graphentheorie ein spezieller Graph, mit dem sich eine Monohierarchie modellieren lässt [Walker90]. Monohierarchie bedeutet, dass ein Knoten im Baum höchstens einen Elternknoten besitzt. Im Gegensatz zur Monohierarchie eines Baumes ist als Namensraum auch eine Polyhierarchie, die durch einen **gerichteten azyklischen Graphen** dargestellt wird, möglich. Polyhierarchie bedeutet, dass ein

Knoten mehr als einen Elternknoten besitzen kann. Ein Graph G wird definiert als ein Paar zweier endlicher Mengen V und E . V bezeichnet dabei die Menge der Knoten und E die Menge der Kanten. Für V und E gelten die Bedingungen:

$V \neq \{\}, V \cap E \neq \{\}$. Jeder Graph besitzt eine auf der Kantenmenge E definierte Abbildung Ψ mit $\Psi: E \mapsto V$. Ein Graph $G = (V, E)$ ist gerichtet, wenn zu jedem $e \in E$ das durch Ψ zugeordnete Paar (v, v') geordnet ist: $v, v' \in V$.

Es existieren die folgenden Anforderungen an einen Namensdienst [Weber00]:

- Eine lange Lebensdauer: Der Namensdienst wird über eine lange Zeit benötigt werden. Während dieser Zeit werden viele Änderungen der Einträge stattfinden.
- Eine hohe Verfügbarkeit: Viele andere Dienste und Programme verlassen sich auf den Namensdienst. Ein nicht verfügbarer Namensdienst macht auch die davon abhängigen Dienste nicht verfügbar.
- Eine Fehlerisolation: Einzelne, lokale Ausfälle dürfen nicht den gesamten Namensdienst stilllegen.
- Eine Toleranz von Misstrauen: Ein großes, offenes System kann keine Komponente haben, der alle Klienten trauen. Namen müssen aber authentisch sein, um sinnvoll zu sein.

Beispiele für Namensdienste, die in der Produktion eingesetzt werden, sind in Anhang B aufgelistet.

3.5.1 Bewertung des Namensdienstes

Am Markt erhältliche Namensdienste spannen typischerweise einen Namensraum mit Baumstruktur auf [Ander07]. Gewünscht ist bei einem Namensdienst in der Produktion aber die Möglichkeit, Namenshierarchien von Anlagen, Geräten, Personen und Objekten direkt so abzubilden, wie sie im Unternehmen anzufinden sind. Dies erleichtert die Wartbarkeit des Namensraums und lässt zu, dass eine direkte Abbildung aller Personen und Objekte eines Unternehmens realitätsgetreu vorgenommen werden kann.

Ein Namensbaum kann in linearer Geschwindigkeit aufgelöst werden, wenn Adjazenzlisten [Diestel00] verwendet werden. Im schlimmsten Fall müssen alle möglichen Pfade zu allen möglichen Knoten betrachtet werden. Dann beträgt die Laufzeit der Tiefensuche [Diestel00]:

$O(V_1 + E_1)$ wobei V_1 für die Anzahl der Knoten und E_1 für die Anzahl der Kanten im Graph steht.

Bei der Suche nach einem Namensraum, der komplexere Beziehungen darstellen kann, muss vor allem auf die Geschwindigkeit bei der Namensauflösung geachtet werden.

3.6 Routenplanung

Im Allgemeinen wird die **Routenplanung** als Prozess der optimalen Wegwahl für Datenpakete vom Sender zum Empfänger über einzelne Netzwerke beschrieben [ITAdmin07]. Die Routenplanung regelt die Nachrichtenübermittlung durch das Netzwerk. Traditionellerweise ist die Routenplanung eine der wesentlichen Aufgaben der OSI-Schicht 3¹¹. Da es sich in dieser Arbeit um eine Nachrichtenorientierte Verteilungsplattform der OSI-Schichten 5-7 handelt, werden nur die Funktionalitäten der Routenplanung auf diesen Schichten näher betrachtet. Die Aufgabe der Routenplanung ist die **Weiterleitung** der Nachricht vom Eingang in die Nachrichtenorientierten Verteilungsplattform bis zum Verlassen dieser. Dabei wird gesteuert, welche Komponenten der Nachrichtenorientierten Verteilungsplattform die Nachricht zugestellt bekommen. Der Namensdienst wird zur Auflösung der Empfangsadresse benötigt, weitere Komponenten sind in Kapitel 3.3 beschrieben. Abbildung 3-15 zeigt den Weg der Nachricht durch die Nachrichtenorientierte Verteilungsplattform.

¹¹ <http://www.it-administrator.de/lexikon/routing.html> (17.07.2007)

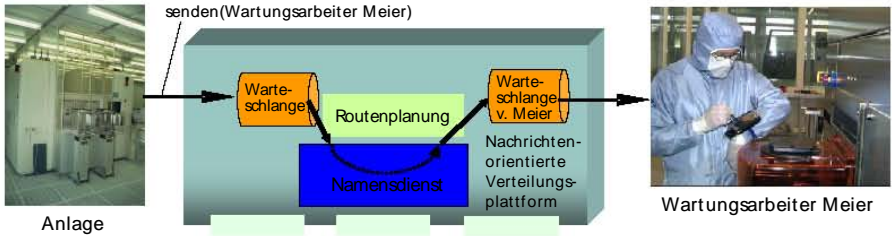


Abbildung 3-15: Nachrichtenweiterleitung innerhalb der Nachrichtenorientierten Verteilungsplattform

Die Anlage sendet eine Nachricht an die Eingangs-Warteschlange der Nachrichtenorientierten Verteilungsplattform mit dem Namen des Empfängers als Empfänger. Der Sender muss den Namen des Empfängers bereits kennen und mit der Nachricht senden, sonst tritt ein Fehler innerhalb der Nachrichtenorientierten Verteilungsplattform auf. Die Nachrichtenorientierte Verteilungsplattform geht davon aus, dass der Sender der Nachricht schon den am besten geeigneten Empfänger gewählt hat und stellt die Nachricht zu, ohne in den Nachrichtenstrom einzugreifen. Es wird das vorhandene Protokoll der Nachrichtenorientierten Verteilungsplattform benutzt, um die Nachricht an die Warteschlange für den Namen des Empfängers weiterzuleiten. Aus der Ausgangs-Warteschlange bekommt der Empfänger dann die Nachricht zugestellt.

In Abbildung 3-16 wurde eine Einteilung in die verschiedenen Routenplanungsverfahren vorgenommen.

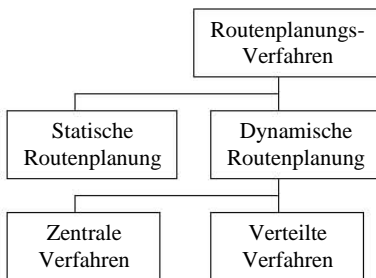


Abbildung 3-16: Routenplanungsverfahren

Routenplanungs-Verfahren lassen sich in eine Statische und Dynamische Routenplanung unterteilen. Die Dynamische Routenplanung unterscheidet weiter in Zentrale Verfahren und Verteilte Verfahren.

Die Routenplanung von Diensten (engl. Service Routing) beschäftigt sich mit der Wegelenkung von Nachrichten innerhalb der Transportschicht und der Schichten einer Verteilungsplattform [Jin03, Jin04]. Es wird zwischen der Punkt-zu-Punkt-Verbindung und dem Gruppenruf [Xu03] unterschieden. Bei der Punkt-zu-Punkt-Verbindung wird weiter zwischen hierarchischer [Jin03] oder verteilter [Xiao02] Routenplanung unterschieden.

Subjektbezogene Systeme beinhalten eine subjektbezogene Routenplanung [Wack97, Castro02, Zhuang01] und inhaltsbezogene Systeme eine inhaltsbezogene Routenplanung [Wang04, Gupta04, Cheung06]. Für die subjektbezogene Routenplanung gibt es die Möglichkeit einer subjektbezogenen Adressierung, die in Produktionssystemen eingesetzt wird. Dabei wird eine Punkt-zu-Punkt Kommunikation aufgebaut, die durch den Namen der Nachricht ein Thema aufbaut und die Nachricht in eine subjektbezogene Nachricht verwandelt [Tibco03].

3.6.1 Bewertung der Routenplanung

Während auf der Ebene der Transportschicht die Routenplanung im Netzwerk gelöst ist, gibt es für die Schichten der Verteilungsplattform Ansätze [Gribble01, Ponne02], die aber zu keinem optimalen Ergebnis führen [Jin04].

Die am Markt verfügbaren Routenplanungskomponenten sind nicht intelligent und flexibel genug, um die Nachricht optimal an die unterschiedlichen Empfänger anzupassen und besitzen keine situationsspezifische Anpassung der Nachricht an den Empfänger. Eine Routenplanung bietet typischerweise keine **Zuordnungsverhandlung** [M2A04]. Dies bedeutet, dass die Routenplanung nicht weiß, ob der Inhalt einer Nachricht auf dem Empfänger ausführbar ist. Wenn beispielsweise die Nachricht einen Microsoft Outlook Termin transportiert, ist es von Vorteil, wenn die Routenplanung weiß, welche Empfangsgeräte das Programm Microsoft Outlook installiert haben und

die Nachricht verwenden können. Je nach momentan verfügbaren Klienten und deren Eigenschaften sollte eine Nachricht möglichst optimal den Empfängern zugestellt werden. Daraus leitet sich die folgende Anforderung an die Routenplanung ab:

- **Einbeziehung aller Informationen des Empfängers für eine situationspezifische Anpassung der Nachricht an das optimale Nachrichtenprotokoll des Empfängers und der Darstellung.**

Es ist nicht immer ausreichend, dass der Sender einer Nachricht immer den Empfänger der Nachricht kennt [M2A03c]. Somit ist ein Empfänger der Nachricht der Nachrichtenorientierten Verteilungsplattform nicht mehr zur Laufzeit zuweisbar. In der Produktion will man oftmals Testnachrichten für Anlagen zur Verfügung stellen, bevor sie an die Nachrichtenorientierte Verteilungsplattform angeschlossen werden. Somit hat man den Vorteil, dass bei der Integration einer Anlage schon Testnachrichten für die Anlage vorhanden sind und man diese nach der Erstellung des Empfängers sofort benutzen kann.

3.7 *Arbeitsablaufverwaltung*

Ein **Arbeitsablauf** wird definiert als die Automatisierung von Geschäftsprozessen, wobei Dokumente, Informationen und Aufgaben von Akteuren nach einem Algorithmus ausgeführt werden [Allen01].

Die Arbeitsablaufverwaltung (engl. Workflow Management, Abk. WfM) umfasst alle Aufgaben zur Ausführung, zur Steuerung, zur Modellierung und zur Spezifikation eines Arbeitsablaufs [Aalst02]. In Abbildung 3-17 ist die Komponente Arbeitsablaufverwaltung innerhalb einer Nachrichtenorientierten Verteilungsplattform abgebildet.

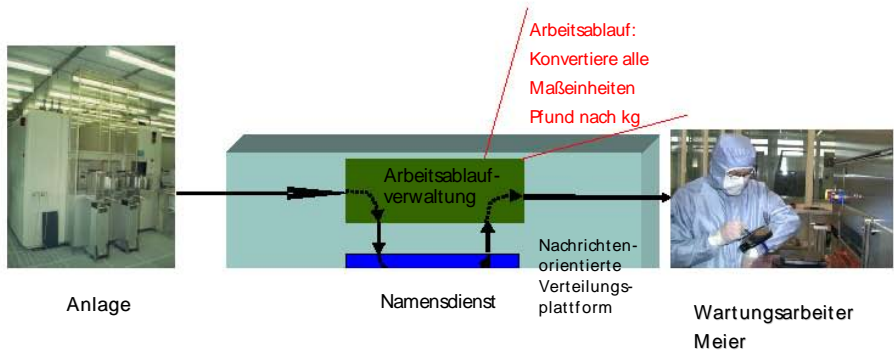


Abbildung 3-17: Nachrichtenübermittlung über die Arbeitsablaufverwaltung

Eine Anlage sendet in diesem Beispiel eine Nachricht mit Wartungsinformationen an Wartungsarbeiter Meier. In dieser Nachricht sind alle Gewichte in Pfund formatiert. Die Applikation von Wartungsarbeiter Meier versteht als Gewichtsangaben nur Kilogramm und ist auf dieses Format angewiesen. Um das Gewichtsformat zwischen Sender und Empfänger in der Nachricht anzupassen, wird in der Arbeitsablaufverwaltung ein Arbeitsablauf ausgeführt, der alle Maßeinheiten von Pfund nach Kilogramm konvertiert.

3.7.1 Bewertung der Arbeitsablaufverwaltung

Um einen Arbeitsablauf ausführen zu können, muss die Nachricht an die Komponente Arbeitsablaufverwaltung weitergeleitet werden. Dies führt wie in folgender Abbildung 3-18 dargestellt zu einer zusätzlichen **Latenzzeit** bei der Nachrichtenübermittlung. Es wird der Weg der Nachricht über die Komponenten der Nachrichtenorientierten Verteilungsplattform in einem Sequenzdiagramm dargestellt. Durch den Aufruf der Komponente Arbeitsablaufverwaltung erhöht sich die Zustellungszeit der Nachricht vom Sender zum Empfänger um diesen Aufruf.

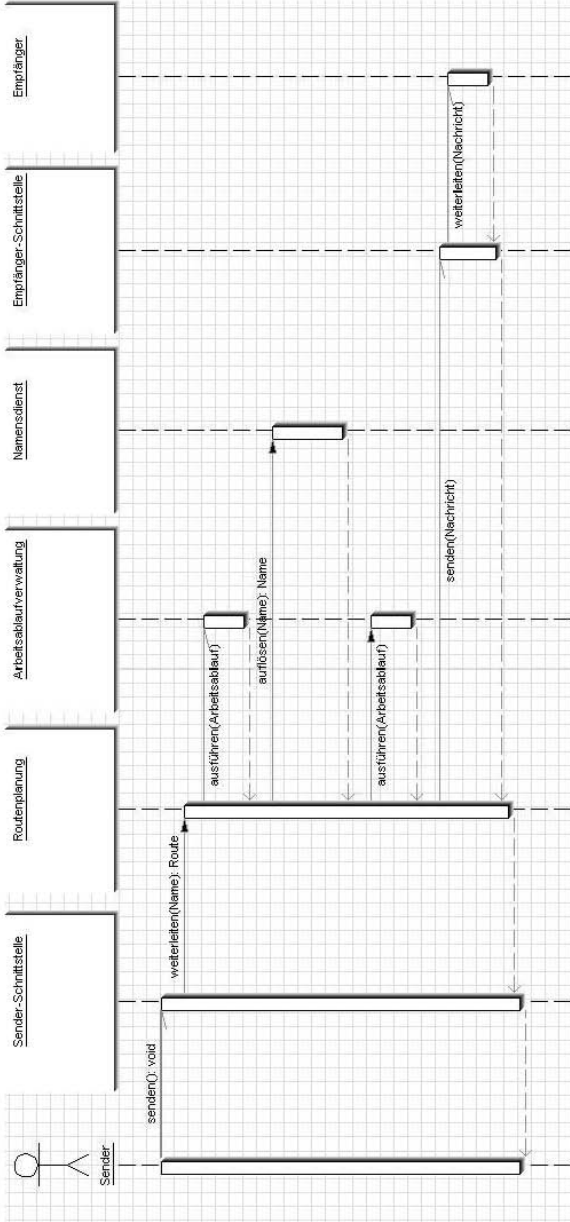


Abbildung 3-18: Sequenzdiagramm der Nachrichtenübermittlung mit Arbeitsablaufverwaltung

Der Sender sendet eine Nachricht durch den Funktionsaufruf „senden(Nachricht)“ an die Sender-Schnittstelle. Von der Sender-Schnittstelle wird die Nachricht an die Nachrichtenorientierte Verteilungsplattform zur Komponente Routenplanung weitergeleitet. Um einen Arbeitsablauf ausführen zu können, wird die Funktion „ausführen(Arbeitsablauf)“ der Komponente Arbeitsablaufverwaltung aufgerufen. Die Routenplanung ruft danach den Namensdienst durch die Funktion „auflösen(Name)“ auf. Um die Nachricht dem Empfänger zuzustellen, wird die Empfänger-Schnittstelle mit der Funktion „senden(Nachricht)“ aufgerufen. Der Empfänger bekommt die Nachricht dann über die Empfänger-Schnittstelle zugestellt. Dieses Beispiel bezieht sich auf die Ausführung eines typischen Arbeitsablaufs innerhalb der Nachrichtenübermittlung. Wenn Sender und Empfänger je mindestens einen Arbeitsablauf ausführen wollen, kommt es zu einem mindestens zweimaligen Aufruf der Arbeitsablaufverwaltung und somit auch zu einer weiteren Erhöhung der Latenzzeit.

Es ist möglich, dass der Sender und Empfänger einer Nachricht jeweils Arbeitsabläufe festlegen, die nicht gemeinsam ausführbar sind und zu Widersprüchen führen. Um das System stabil zu halten und eine möglichst große Anzahl von Arbeitsabläufen ausführen zu können, sollte es eine **Überwachung** der Arbeitsabläufe geben. Die Ausführung der Arbeitsabläufe muss in diesem Fall durch **Regeln** oder **Verhandlungen** abgestimmt werden, damit das Gesamtsystem keine Fehler aufweist. Widersprüchliche Arbeitsabläufe sollten situationsspezifisch mit dem momentanen Zustand des Gesamtsystems abgestimmt und aufgelöst werden. Daraus leiten sich die Anforderungen

- **Auflösung/Verhandlung von konkurrierenden Arbeitsabläufen** und
- **Situationsspezifische Abstimmung von Arbeitsabläufen**

ab.

3.8 Nachrichtenzustellung in der Wartung

Die Wartung spielt durch das Instandhalten und Verbessern der Verfügbarkeit, der Produktqualität, der Sicherheitsanforderungen, der Produktionskosten und der

Globalisierung in der Fertigung eine immer größere Rolle [Al03]. Durch die Einführung von Internet und mobilen Geräten in der Produktion verändern sich die Fertigungsabläufe von der Philosophie der Fabrikintegration zur Philosophie von elektronischer Fertigung und elektronischen Lieferketten. Dies transformiert die Fertigung von einer lokalen Fertigungsautomatisierung zu einer globalen Firmen und Geschäftsprozessautomatisierung.

In der Wartung gibt es drei grundlegende Ausrichtungen: Korrigierende Wartung, Geplante Wartung und Vorbeugende Wartung. Die Wartung wird in Kooperation mit Wartungsarbeitern, Wartungstechnikern und Ingenieuren durchgeführt. Wartungsarbeiter müssen in der Lage sein das Produktionssystem zu überwachen, analysieren, vergleichen, neu konfigurieren und aufrechterhalten. Informationszugriffe in Echtzeit sind notwendig, um die Fertigungseffizienz zu steigern.

Die Kommunikation spielt in der Wartung eine wichtige Rolle, da Wartungsnachrichten versendet werden müssen [Desfo06]. Zum einen müssen die Befehle von Aktoren wie dem Anlagensteuerungsrechner, Zellenrechner oder Überwachungsprogramm möglichst schnell versendet werden. Zum anderen müssen Statusnachrichten von Anlagen und des Gesamtsystems für Wartungsarbeiter sofort verfügbar sein.

3.8.1 *Bewertung der Nachrichtenzustellung in der Wartung*

Der Stand der Technik der Nachrichtenzustellung in der Wartung ist momentan noch weit entfernt von der Vollautomatisierung [Han06]. Aufgrund der fehlenden Informationsverbreitung kann eine Diagnose und Reparatur an defekten Anlagen normalerweise nicht sofort durchgeführt werden und verursacht deshalb lange Ausfallzeiten und Produktivitätsverlust. Um die Informationsverbreitung möglichst effektiv in der Produktion einzusetzen, muss schon bei der Konzeption einer Verteilungsplattform darauf geachtet werden, dass Wartungsarbeiter Nachrichten zu jeder Zeit an jedem Ort in genau dem richtigen Format erhalten können.

3.9 Zusammenfassung der Anforderungen

Aus den vorausgegangenen Bewertungen im Kapitel Stand der Technik lassen sich einige Anforderungen zusammenfassen und daraus Entwicklungsschwerpunkte ableiten. Als Verteilungsplattform für die Produktion ist, aufgrund der hohen Anzahl von Klienten und Nachrichten die Anforderung an einen hohen Nachrichtendurchsatz unverzichtbar:

- **Punkt-zu-Punkt Nachrichtenübermittlung mit höherem Durchsatz als bei bestehenden Systemen.**

Gewünscht ist bei einem Namensdienst in der Produktion die Möglichkeit, Namenshierarchien von Anlagen, Geräten, Personen und Objekten direkt abzubilden, wie sie im Unternehmen anzufinden sind mit der folgenden Anforderung:

- **Abbildung von komplexen Beziehungen im Namensraum.**

Je nach momentan verfügbaren Klienten und deren Eigenschaften sollte eine Nachricht möglichst optimal den Empfängern zugestellt werden. Daraus leitet sich die folgende Anforderung an die Routenplanung ab:

- **Einbeziehung aller Informationen des Empfängers für eine situationsspezifische Anpassung der Nachricht an das optimale Nachrichtenprotokoll des Empfängers und der Darstellung.**

Eine grundlegende Anforderung für eine flexible und hochdynamische Nachrichtenübermittlung ist, dass eine schnelle Möglichkeit zur Ausführung von Arbeitsabläufen für Sender und Empfänger existieren soll:

- **Schnelle Ausführung von Arbeitsabläufen von Sender und Empfänger.**

Die Ausführung der Arbeitsabläufe muss in diesem Fall durch **Regeln** oder **Verhandlungen** abgestimmt werden, damit das Gesamtsystem keine Fehler aufweist. Widersprüchliche Arbeitsabläufe sollten situationsspezifisch mit dem momentanen Zustand

des Gesamtsystems abgestimmt und aufgelöst werden. Daraus leiten sich die Anforderungen:

- **Auflösung/Verhandlung von konkurrierenden Arbeitsabläufen** und
- **Situationspezifische Abstimmung von Arbeitsabläufen**

ab.

3.10 Bewertung der abgeleiteten Anforderungen einer Nachrichtenorientierten Verteilungsplattform für die Produktion

Die Anforderungen an eine Nachrichtenorientierte Verteilungsplattform in der Produktion sind innerhalb einer Marktrecherche [Knoll06a] bei 16 der erfolgreichsten Verteilungsplattformen untersucht worden:

- TIBCO Rendezvous¹²
- IIT Software SwiftMQ¹³
- Fiorano MQ 5 / MQ Server Components¹⁴
- Sonic Software SoniqMQ¹⁵
- Spiritsoft Spiritwave
- Pramati Pramati Server¹⁶
- BEA messageQ¹⁷
- Sun Java System Message Queue¹⁸
- IBM WebSphere MQ¹⁹
- PCB Systems Nirvana²⁰
- IONA Artix²¹

¹² <http://www.tibco.com/software/messaging/rendezvous/default.jsp> (17.07.2007)

¹³ <http://www.swiftmq.com/index.html> (17.07.2007)

¹⁴ <http://www.fiorano.com/products/fmq/overview.htm> (17.07.2007)

¹⁵ <http://www.sonicsoftware.com/products/sonicmq/index.ssp> (17.07.2007)

¹⁶ http://www.pramati.com/index.jsp?id=pro_psv41 (17.07.2007)

¹⁷ <http://e-docs.bea.com/tuxedo/msgq/> (17.07.2007)

¹⁸ http://www.sun.com/software/products/message_queue/index.xml (17.07.2007)

¹⁹ http://www-950.ibm.com/ecatalog/Detail.wss?locale=de_DE&synkey=G106020C87422R91 (17.07.2007)

²⁰ <http://www.nirvana.com/> (17.07.2007)

- NetMotion Mobility XE²²
- Spread Concepts LLC Spread Toolkit²³
- Object Web JORAM²⁴
- JBoss JBoss MQ²⁵
- The OpenJMS Group OpenJMS²⁶

Es wurde ein Rückantwortdokument (engl. Request for Evaluation, Abk. RfE) erstellt, dass von den Anbietern der Produkte ausgefüllt wurde. Der RfE wurde durch eine weitere Analyse anhand von Test-Installationen, Datenblättern sowie Produktbeschreibungen detailliert. Abbildung 3-19 zeigt das namensneutralisierte Ergebnis des Rückantwortdokuments mit einer Bewertungszahl als Ergebnis.

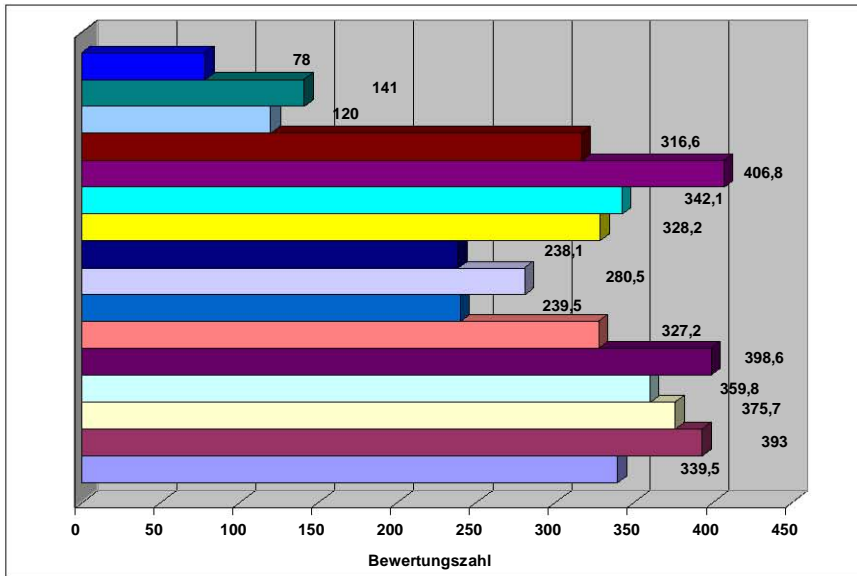


Abbildung 3-19: Ergebnis des Rückantwortdokuments

²¹ <http://www.iona.com/products/artix/> (17.07.2007)

²² <http://www.netmotionwireless.com/products/> (17.07.2007)

²³ <http://www.spread.org/index.html> (17.07.2007)

²⁴ <http://joram.objectweb.org/> (17.07.2007)

²⁵ <http://www.jboss.org/wiki/Wiki.jsp?page=JBossMQ> (17.07.2007)

²⁶ <http://openjms.sourceforge.net/> (17.07.2007)

Es konnte maximal eine Bewertungszahl von 600 erreicht werden. Die Nachrichtenorientierte Verteilungsplattform mit der höchsten Bewertungszahl kommt gerade einmal auf 407 Punkte. Mit diesem Ergebnis wird deutlich, dass nur 2/3 der gewünschten Funktionalitäten und Anforderungen durch bestehende Systeme erfüllt werden. Das Ergebnis dieser Marktrecherche ist, dass keine untersuchte Verteilungsplattform alle Anforderungen aus Kapitel 3.9 erfüllt. Innerhalb dieser Arbeit wurde diese Marktrecherche aufgegriffen und auf 66 Nachrichtenorientierte Verteilungsplattformen ausgeweitet, die in Anhang C aufgelistet sind. Nach der Ausweitung der Marktstudie wurde darunter ebenfalls keine Nachrichtenorientierte Verteilungsplattform identifiziert, die alle abgeleiteten Anforderungen aus Kapitel 3.9 erfüllt.

4 Konzeption und Entwicklung

In Kapitel 3 wurde als Stand der Wissenschaft und Technik beschrieben, dass eine Nachrichtenorientierte Verteilungsplattform typischerweise aus den Komponenten Namensdienst, Routenplanung, Arbeitsablaufverwaltung, Sicherheit, Datenbankverbindung, Transaktionsdienst und Protokollverwaltung besteht.

In den folgenden Unterkapiteln wird detailliert auf die Konzeption und Neuentwicklung der Komponenten Namensdienst, Routenplanung und Arbeitsablaufabstimmung eingegangen. Im letzten Unterkapitel wird die Integration der Komponenten in die Nachrichtenorientierte Verteilungsplattform betrachtet.

4.1 Konzeption

Um die **Anforderungen** aus Kapitel 3.9 zu erfüllen, wurde ein Gesamtkonzept für die Basiskomponenten einer Nachrichtenorientierten Verteilungsplattform erstellt. Die Anforderungen führen zu Konzeptionsentscheidungen bei der Wahl der zu implementierenden Komponenten Namensdienst, Routenplanung, Arbeitsablaufverwaltung und Arbeitsablaufabstimmung.

Eine innovative **Konzeptionsentscheidung** wurde getroffen um eine schnelle Nachrichtenübermittlung und zugleich eine hochdynamische Nachrichtenzustellung zu ermöglichen. Dieses Konzept sieht vor, dass der **Namensdienst** um eine Ausführung von Arbeitsabläufen **erweitert** wird, die die Nachrichtenzustellung betreffen. Dadurch wird für diesen Fall **keine Arbeitsablaufverwaltungs-Komponente** mehr benötigt und die Latenzzeit für deren Aufruf eingespart. Dementsprechend wird der Namensdienst durch neue Objekte und Klassen angereichert, die die Aufgabe der Arbeitsablaufverwaltung in schneller Geschwindigkeit übernehmen. Damit die Ausführung von Arbeitsabläufen nicht das Gesamtsystem beeinträchtigt und nicht vereinbare Arbeitsabläufe aufgelöst werden können, wird eine neue Komponente, die **Arbeitsablaufabstimmung** eingeführt. Die Transformation von der Konzeption bestehender Systeme zu

der Konzeption in dieser Arbeit ist in Abbildung 4-1 vereinfacht dargestellt und wird in Kapitel 4.4 weiter detailliert.

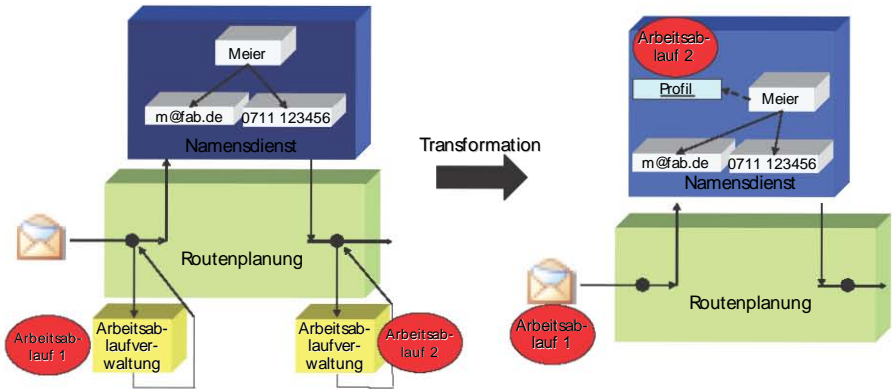


Abbildung 4-1: Transformation der Arbeitsablaufverwaltung in den Namensdienst

Die Innovation der neuen Konzeption liegt darin, dass die Komponente Arbeitsablaufverwaltung für Arbeitsabläufe der Nachrichtenzustellung nicht mehr benötigt wird. Die Funktionalität wird in den Namensdienst und die Nachricht übertragen, sodass durch das Gesamtkonzept eine schnellere Ausführung und Auflösung von Arbeitsabläufen möglich wird.

4.1.1 Nachrichtenübermittlung

Die Anforderung einer Punkt-zu-Punkt Nachrichtenübermittlung mit höherem Durchsatz als bei bestehenden Systemen wird durch verschiedene Faktoren beeinflusst, wie z.B. die Konfiguration der Nachrichtenorientierten Verteilungsplattform. Innerhalb einer Nachrichtenorientierten Verteilungsplattform gibt es verschiedene Möglichkeiten, die **Warteschlangen** für die Punkt-zu-Punkt Nachrichtenübermittlung zu konfigurieren. Die in Abbildung 4-2 dargestellte **Konfiguration** besitzt nach den durchgeführten Laufzeittests in Kapitel 6.1 den höchsten Nachrichtendurchsatz und wurde deshalb für diese Arbeit übernommen.

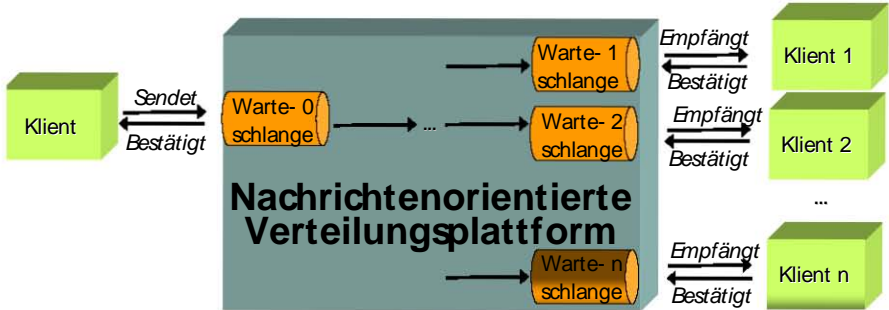


Abbildung 4-2: Konfiguration der Warteschlangen innerhalb der Nachrichtenorientierten Verteilungsplattform

Der Klient sendet die Nachricht immer direkt an die Warteschlange 0 der Nachrichtenorientierten Verteilungsplattform. Dies ist die Eingangs-Warteschlange für Nachrichten aller Klienten im System. Damit wird zum einen ein einheitlicher Zugangspunkt für die Nachrichtenorientierten Verteilungsplattform definiert und zum anderen kann intern auf eingehende Nachrichten standardisiert auf eine Warteschlange reagiert werden. Diese Konfiguration stellt nur einen Geschwindigkeitsgewinn dar, wenn die Eingangs-Warteschlange nicht überfüllt und somit nicht zu einem Engpass im System wird. Um zu verhindern, dass diese oder irgendeine andere Warteschlange einen Engpass darstellt, wurde eine Überwachung von Warteschlangen als Funktionalität ergänzt. Eine Überwachungskomponente zählt die Anzahl der Nachrichten in jeder Warteschlange. Jede Warteschlange, die bis zu einem bestimmten, frei wählbaren Prozentsatz gefüllt ist, löst eine Warnmeldung aus. Dadurch ist es möglich, Warteschlangen, die an ihre Kapazitätsgrenze stoßen und Warteschlangen, die die Geschwindigkeit des Gesamtsystems negativ beeinflussen, zu erkennen. Sobald eine Warteschlange an der Kapazitätsgrenze erkannt wird, wird eine weitere Eingangs-Warteschlange erstellt, die solange Nachrichten empfängt, bis ein Normalzustand bei der ursprünglichen Eingangs-Warteschlange wiederhergestellt ist. In dieser Arbeit wurde das Kriterium zum Erstellen einer neuen Warteschlange auf 85% der maximalen Anzahl von Nachrichten in der Warteschlange festgelegt. Abbildung 4-3 zeigt eine überfüllte Eingangs-Warteschlange 1. Eine neue Eingangs-Warteschlange 2 wird erstellt und alle zukünftigen Klienten verbinden sich daraufhin automatisch mit Eingangs-Warteschlange 2.

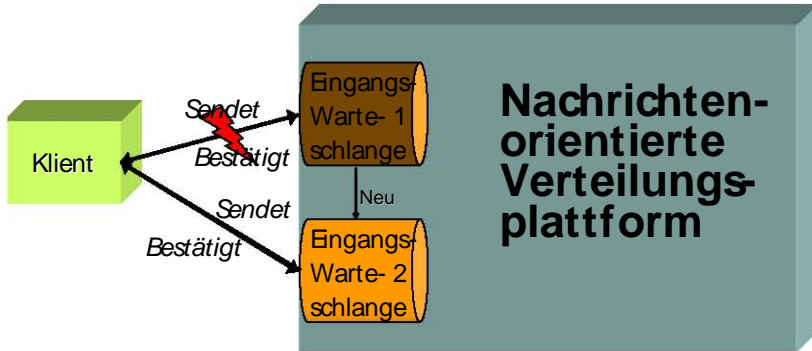


Abbildung 4-3: Replikation einer Eingangswarteschlange

Für die Anzahl der Warteschlangen, die ein Server verwalten kann, gibt es eine Begrenzung. Diese Begrenzung ist abhängig von der zugrunde liegenden Transportschicht auf OSI-Schicht 4 und liegt bei der hier betrachteten Nachrichtenorientierten Verteilungsplattform bei 256 Warteschlangen.

Durch die im vorigen Kapitel 3.3.1 dargestellten Vorteile einer Verteilten Softwarearchitektur wurde ein Verteiltes System als Basis für die Nachrichtenorientierte Verteilungsplattform ausgewählt. Dadurch muss darauf geachtet werden, dass Server repliziert werden, bevor die Anzahl der Warteschlangen eines Servers die Systemgrenze überschreitet. Durch die Replikation passt sich das System dynamisch an die Umgebung an und verhilft dadurch zu einer besseren Performanz.

Für die Nachrichtenübermittlung gibt es verschiedene Arten von Nachrichten innerhalb der Nachrichtenorientierten Verteilungsplattform. Somit kann schneller bestimmt werden, wie die Nachricht verarbeitet werden muss. Die unterschiedlichen Typen sind:

- Punkt-zu-Punkt-Nachricht
- Registrierungs-Nachricht
- Deregistrierungs-Nachricht

Typischerweise werden innerhalb der Nachrichtenorientierten Verteilungsplattform Punkt-zu-Punkt-Nachrichten versendet. Die Registrierungs- und Deregistrierungs-Nachrichten werden eingesetzt, um Klienten am Namensdienst an- und abzumelden.

Der Aufbau der **Nachricht** wird detailliert beschrieben, da dadurch die Routenplanung die Nachricht schnell weiterleiten kann. Außerdem ist es möglich, Arbeitsabläufe zur Nachrichtenübermittlung mit der Nachricht zu versenden, die dann innerhalb der Nachrichtenorientierten Verteilungsplattform ausgeführt werden. In Abbildung 4-4 ist der Aufbau der Nachricht in der Unified Modelling Language (UML) [Kru99, West01] zu sehen.

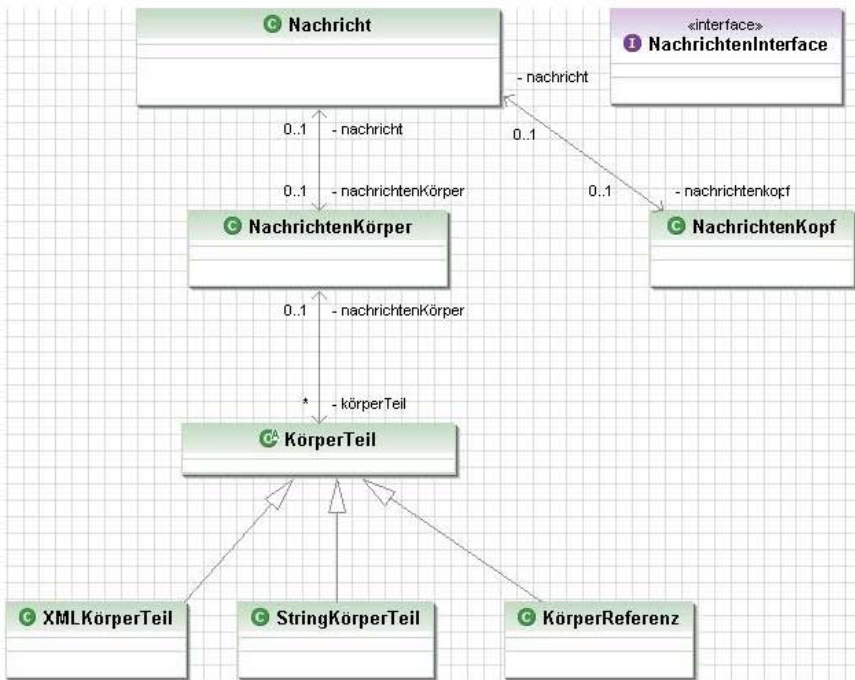


Abbildung 4-4: UML Diagramm der Nachricht

Die Nachricht wird immer über das NachrichtenInterface von den Komponenten der Nachrichtenorientierten Verteilungsplattform aufgerufen. Eine Nachricht besteht aus einem NachrichtenKopf und einem NachrichtenKörper. Der Nachrichtenkopf beinhaltet alle Informationen über die Nachricht, die während der Nachrichtenzustellung benötigt werden. Diese Informationen können schnell abgefragt werden, ohne dass die Nachricht gelesen werden muss und sind im Folgenden dargestellt:

- Nachrichten-ID
- Sender
- Empfänger
- Route
- Zeitstempel
- Verfallszeit
- Dienstgüte
- Korrelations-ID
- Arbeitsablauf-ID

Die Nachrichten-ID sorgt für eine eindeutige Identifizierung der Nachricht. Als Empfänger können beliebig viele Namen, Adressen oder Themen angegeben werden. Durch die Route kann der Empfänger der Nachricht die zusätzliche Information mitgeben, welche Komponenten innerhalb der Nachrichtenorientierten Verteilungsplattform die Nachricht erhalten sollen. Mit dem Zeitstempel wird der Zeitpunkt des Nachrichtenversands festgelegt. Durch die Verfallszeit wird festgelegt, wie lange die Nachricht im System aktiv sein soll, bevor sie gelöscht wird. Die Nachricht wird gelöscht, wenn $\text{Verfallszeit} = \text{Systemzeit} - \text{Zeitstempel}$ eintritt. Innerhalb der Dienstgüte werden die Parameter Sicherheit, Zustellungszuverlässigkeit, Priorität und Protokollierung festgelegt. Die Korrelations-ID, ordnet die Nachrichten zu, sodass die Antwort auf eine Nachricht bestimmt werden kann. Innerhalb der Arbeitsablauf-ID werden alle Arbeitsabläufe, die vom Sender festgelegt werden identifiziert.

Der Nachrichtenkörper besteht aus beliebig vielen KörperTeilen. Ein KörperTeil ist eine abstrakte Klasse und wird durch die Klassen XMLKörperTeil, StringKörperTeil oder KörperReferenz realisiert.

Die Erfüllung der Anforderung einer **schnelleren Nachrichtenübermittlung** ist durch verschiedene Konzeptionsansätze denkbar. Ein Ansatz ist, die anfallenden Daten für eine Nachricht zu komprimieren, sodass die Nachricht dadurch kleiner wird und schneller übermittelt werden kann. Da sich aber die Nachrichtengröße nicht linear zur Geschwindigkeit der Nachrichtenübermittlung verhält, sondern nur minimale Auswirkungen zeigt [Stiedl04, Knoll06a], führt dieser Ansatz nur bedingt zum Erfolg. Eine weitere Möglichkeit, eine schnellere Nachrichtenübermittlung zu erreichen wäre,

die Daten für den Inhalt einer Nachricht schon beim Sender zu filtern und dadurch weniger Nachrichten mit kleinerem Inhalt zu versenden. Dieser Ansatz wird innerhalb von E-Diagnostics [Wohl01] schon verfolgt und wird deshalb in dieser Arbeit nicht näher untersucht.

Der Ansatz dieser Arbeit ist es, durch eine neues Konzeptionsmuster die Basiskomponenten einer Nachrichtenorientierten Verteilungsplattform so zu gestalten, dass in einem geschlossenen Ansatz die Geschwindigkeit und Flexibilität erhöht wird. Dies wird hauptsächlich durch die neue Konzeption des Namensdiensts und der Routenplanung, die gleichzeitig die Aufgaben der Arbeitsablaufverwaltung übernehmen, erreicht. Dieser Ansatz wurde gewählt, da er eine Neuerung darstellt und mit den beiden oben genannten Ansätzen kombiniert werden kann.

Außerdem tragen die in diesem Kapitel vorgestellten Konzepte der Nachrichtenübermittlung zu einer schnelleren Nachrichtenzustellung bei durch:

- Konfiguration der Warteschlangen innerhalb der Nachrichtenorientierten Verteilungsplattform.
- Festlegung verschiedener Nachrichtentypen, um spezielle Aufgaben schneller zu erkennen und abzuarbeiten.
- Nachrichtenstruktur inklusive Arbeitsabläufen spart die Latenzzeit ein, die beim Aufruf einer Arbeitsablaufverwaltung entsteht.

4.1.2 Namensdienst

Eine direkte Abbildung von komplexen Beziehungen im Namensraum wird durch einen Graph als **Namensraum** erreicht, liefert aber ein langsames Laufzeitverhalten [Diestel00]. Da komplexe Graphen als Namensraum hohe Auflösungszeiten benötigen, wurde ein spezieller Graph, der gerichtete, azyklische Graph ausgewählt.

Sobald ein Graph azyklisch ist, ist die Tiefen- oder Breitensuche vollständig und ergibt im schlechtesten Fall ein lineares Laufzeitverhalten, ähnlich wie in einem Baum $O(2 \cdot V + 2 \cdot E)$ wobei V für die Anzahl der Knoten und E für die Anzahl der Kanten

im Graph steht, wenn doppelt verkettete Adjazenzlisten verwendet werden. Somit lässt sich die Namensauflösung in einem gerichteten, azyklischen Graph schnell durchführen und wird für die Entwicklung übernommen. Mit einem azyklischen Graphen lassen sich alle bisher in der Produktion vorhandenen Hierarchien von Personal, Anlagen und Material abbilden. Der Namensraum wird innerhalb der Komponente Namensdienst entwickelt.

Der Namensdienst wird von Grund auf neu konstruiert, speziell ausgelegt auf die Anforderung einer schnelleren Nachrichtenübermittlung. Bestehende Namensdienste konnten nicht wieder verwendet und modifiziert werden, da diese Komponente komplett eigene Objekte und Objekttypen verwendet, die zusätzlich Arbeitsabläufe beinhalten und auf einer eigens konzeptionierten Datenstruktur auf hohen Durchsatz von Nachrichten und Flexibilität optimiert wurde.

4.1.3 Routenplanung

Um eine Nachrichtenumwandlung in das Protokoll des Empfängers durchführen zu können, muss dieses Empfangsprotokoll der Nachrichtenorientierten Verteilungsplattform bekannt sein. Zur Laufzeit sammelt die entwickelte Komponente Routenplanung die Informationen des Nachrichtenempfängers in einer Zuordnungstabelle ein und wandelt die Nachricht, unter Zuhilfenahme der Protokollverwaltung und des Protokollumsetzers, in das optimale Format für den Empfänger. Die Routenplanungskomponente überprüft zur Laufzeit, welche Klienten mit der Nachrichtenorientierten Verteilungsplattform verbunden sind und welche Eigenschaften diese Klienten haben. Die Erfüllung der Anforderung der Einbeziehung aller Informationen des Empfängers für eine situationsspezifische Anpassung der Nachricht an das optimale Nachrichtenprotokoll des Empfängers und der Darstellung ergibt sich durch:

- Konzeption einer Zuordnungstabelle, die alle Informationen der verbundenen Klienten enthält.
- Situationsspezifischer Abgleich des Gesamtsystems zur Laufzeit.

4.1.4 Arbeitsablaufverwaltung

Die Ausführung von Arbeitsabläufen findet in Nachrichtenorientierten Verteilungsplattformen normalerweise in der Komponente Arbeitsablaufverwaltung statt, die mit einer zusätzlichen Latenzzeit aufgerufen werden muss. Da aber gleichzeitig die Anforderung für eine schnellere Nachrichtenübermittlung gestellt ist, wird in der Konzeption diese Komponente nicht berücksichtigt, sondern Arbeitsabläufe für die Nachrichtenzustellung werden direkt im Namensdienst ausgeführt. Dieses neue Konzept sorgt dafür, dass Arbeitsabläufe für die Nachrichtenübermittlung keine Arbeitsablaufverwaltung benötigen und innerhalb des Namensdienstes, der bei jeder Nachrichtenzustellung aufgerufen wird, ausgeführt werden. Somit wird die Latenzzeit eingespart und die Anforderung Schnelle Ausführung von Arbeitsabläufen von Sender und Empfänger wird erfüllt durch:

- Ausführung von Arbeitsabläufen im Namensdienst spart Latenzzeit, die sonst beim Aufruf einer Arbeitsablaufverwaltung entsteht.

Innerhalb der Entwicklung des Namensdienstes und der Arbeitsablaufabstimmung wird gezeigt, dass diese Anforderung erfüllt wird.

4.2 Entwicklung des Namensdienstes

Da am Markt verfügbare Namensdienste nur Baumstrukturen und nicht Graphen als Namensraum anbieten, wie in Kapitel 3.5 beschrieben wurde, beruht die Entwicklung auf einer kompletten **Neumodellierung** und Implementierung der Komponente Namensdienst. Dies ist wichtig, da sonst komplexe Beziehungen im Namensraum, wie in Kapitel 3.5.1 ersichtlich, nicht abgebildet werden können.

Der Namensraum als azyklischer Graph wurde in der Universal Modelling Language modelliert. Um den Namensraum zu erstellen, müssen zuerst die logischen Abhängigkeiten der Objekte des Namensraums aufgebaut werden. Zum Aufbau des Namensraums wurden eigene Objekte und Objekttypen entworfen. Abbildung 4-5 zeigt die Objekte des Namensraums und deren Abhängigkeiten.

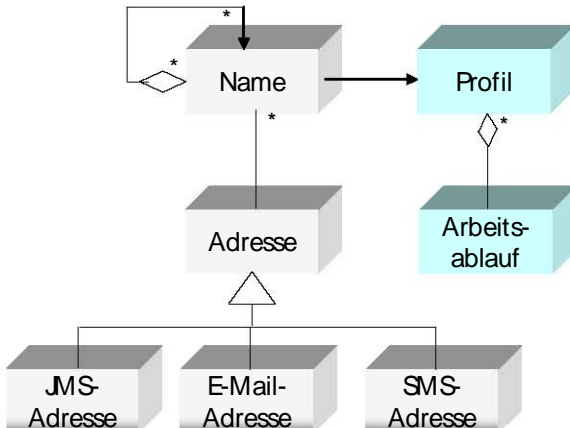


Abbildung 4-5: UML Modell des azyklischen Namensgraph

Das Objekt **Name** besteht aus dem zu beschreibenden Namen für eine Person, eine Anlage, ein Gerät, eine Anwendung, eine Gruppe oder eine Rolle, sowie einer Verknüpfung zu Eltern und Kindern als Hash Set. Das Wurzelement eines Namensgraphen besteht immer aus einem Namen. Dieses Wurzelement besitzt ein Alleinstellungsmerkmal indem es keine Verknüpfung zu einem Elternobjekt besitzt. Ein Name kann wiederum beliebig viele Namen als Kinder besitzen. Somit wird über ein Kompositionsmuster ein azyklischer Graph aufgebaut. Der azyklische Graph wird dadurch aufgebaut, dass mehrere Eltern zulässig sind. Jeder Name, der keine weiteren Namen als Kinder besitzt, wird mit dem Objekt **Adresse** verknüpft. Eine Adresse stellt einen abstrakten Objekttypen dar. Die Realisierung einer Adresse besteht aus verschiedenen Adresstypen wie JMS-Adresse, E-Mail-Adresse, SMS-Adresse, Voice-Over-IP-Adresse oder beliebige weitere Adresstypen, die nachträglich frei entworfen werden können. Für jedes Kommunikationsprotokoll kann eine Adressklasse angegeben werden, die dann durch das Adressobjekt definiert wird. Ein Adressobjekt ist immer mit mindestens einem Namensobjekt verbunden. Ein konkreter Adresstyp stellt immer einen Eckknoten in dem azyklischen Graphen dar und besitzt somit keine Verknüpfung zu Kindern. Eine Adresse als Eckknoten ist bei der Namensauflösung notwendig, damit eine konkrete physikalische Adresse eines Klienten zurückgeliefert werden kann.

Jeder Name kann eine Verknüpfung zu einem **Profil** besitzen. Ein Profil ist ein abstraktes Objekt, das aus verschiedenen konkreten Arbeitsabläufen bestehen kann. Die Arbeitsabläufe beinhalten Zustellungsregeln für die Nachrichtenübermittlung. Typischerweise ist in einem Profil die Information enthalten, welche Kindknoten eines Namensobjekts aufgelöst werden sollen. Durch Profile können Arbeitsabläufe für die Nachrichtenzustellung an verschiedene Empfänger zugeordnet werden. Abbildung 4-6 zeigt beispielhaft einen Namensgraph mit einem Profil an jedem Namen.

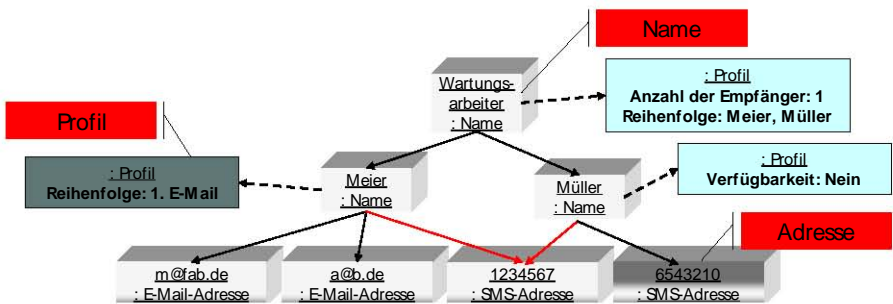


Abbildung 4-6: Namensgraph mit Profilen

In diesem Beispiel legt das Profil am Namen **Wartungsarbeiter** fest, dass durch den Arbeitsablauf in der ersten Zeile „Anzahl der Empfänger: 1“ bei einer Adressierung von **Wartungsarbeiter** immer nur ein Kind, d.h. eine Person die Nachricht erhält. Beliebige weitere Einstellungen sind die Auflösung aller oder nCN Personen unter dem Namensobjekt **Wartungsarbeiter**. Im vorliegenden Profil ist in der zweiten Zeile ein zusätzlicher Arbeitsablauf „Reihenfolge: Meier, Müller“ definiert, der die Reihenfolge der aufzulösenden **Wartungsarbeiter** angibt. Dadurch soll immer zuerst **Wartungsarbeiter Meier** die Nachricht erhalten, als zweiter **Wartungsarbeiter Müller**. Somit wird bei einer Verfügbarkeit von **Meier**, immer nur **Wartungsarbeiter Meier** eine Nachricht, die an die Gruppe **Wartungsarbeiter** adressiert ist, zugestellt. Es ist auch möglich, wie im Profil von **Wartungsarbeiter Müller**, mit der Zeile „Verfügbarkeit: Nein“ eine Abwesenheit anzugeben. Dadurch wird erkenntlich gemacht, dass man momentan keine Möglichkeit hat Nachrichten zu empfangen. Nachrichten an die Gruppe **Wartungsarbeiter** werden zuerst an verfügbare **Wartungsarbeiter** weitergeleitet. Das Profil von **Meier** gibt mit dem Arbeitsablauf „Reihenfolge: 1. E-Mail“ an, dass ihm zuerst immer eine E-

Mail als erste Priorität zugestellt werden soll. Wenn keine Anzahl von Empfängern angegeben ist, wie im Profil von Meier, dann leitet die Standard Konfiguration Nachrichten an alle Kinder eines Namensobjektes weiter und es werden alle Kinder aufgelöst. In diesem Fall bekommt Meier die Nachricht über alle drei eingetragenen Adressen zugestellt.

Somit erfüllt der Namensdienst die Anforderung der Abbildung von komplexen Beziehungen im Namensraum durch:

- Darstellung des Namensraums als gerichteter, azyklischer Graph.

Auf die entwickelte Objektstruktur des Namensraums kann direkt mit Hilfe der Funktion „resolveName(Name name)“ zugegriffen werden. Im Gegensatz zu Namensdiensten, die erst eine Zugriffsklasse (engl. factory) erstellen müssen, danach eine Steuerungsklasse (engl. handler) auf der Zugriffsklasse anlegen und dann erst den Zugriff auf den Namensdienst gewähren, ergibt sich eine schnellere Namensauflösung. Das schlanke Design des Namensdienstes und dadurch eine leichtgewichtige Architektur der Klassen und Objekte des Namensraums, führen zu einer schnelleren Namensauflösung. Besonders bei der Ausführung von Arbeitsabläufen innerhalb der Namensauflösung wird die Latenzzeit eingespart, die die Arbeitsablaufverwaltungs-Komponente zusätzlich verursacht. Da die Namensauflösung wesentlicher Bestandteil der Nachrichtenzustellung ist, wird ebenfalls zu der Anforderung:

- Punkt-zu-Punkt Nachrichtenübermittlung mit höherem Durchsatz als bei bestehenden Systemen.

beigetragen.

Durch das Objekt Profil und der Möglichkeit, Arbeitsabläufe in einem Profil zu definieren, trägt der Namensdienst zum Erreichen der Anforderung:

- Schnelle Ausführung von Arbeitsabläufen von Sender und Empfänger.

bei.

4.3 Entwicklung der Routenplanung

Die Routenplanung ist die zentrale Komponente dieser Arbeit. Hier werden alle anderen Komponenten der Nachrichtenorientierten Verteilungsplattform verwaltet und gesteuert. Innerhalb der Nachrichtenorientierten Verteilungsplattform ist die Routenplanung die erste Komponente, die eine Nachricht von einem Klienten entgegennimmt.

Abbildung 4-7 zeigt auf, welche Komponenten innerhalb der Nachrichtenorientierten Verteilungsplattform bei der Nachrichtenübermittlung mindestens durchlaufen werden.

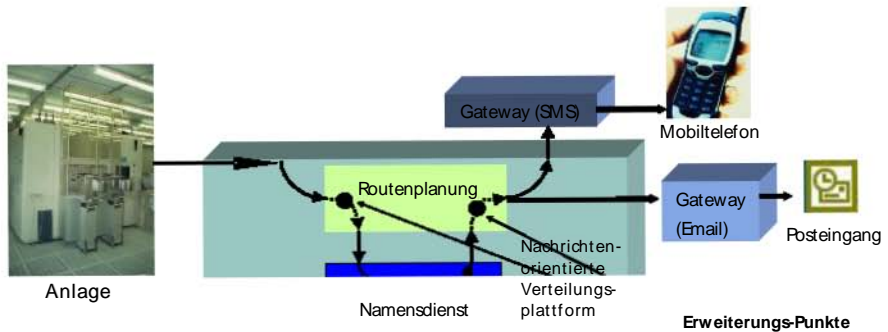


Abbildung 4-7: Sequenz der Nachrichtenübermittlung im Komponentendiagramm

Eine Anlage sendet eine Nachricht an die Gruppe Wartungsarbeiter. Die Nachricht wird an die Nachrichtenorientierte Verteilungsplattform gesendet. Die Routenplanung nimmt diese Nachricht entgegen und analysiert den Kopf der Nachricht. Durch die Analyse des Nachrichtenkopfes und der Art der Nachricht wird entschieden, welche weiteren Komponenten die Nachricht durchlaufen muss. An den Erweiterungs-Punkten kann die Nachricht an weitere Komponenten übergeben werden. Einige Komponenten werden am Erweiterungs-Punkt vor der Namensauflösung im Namensdienst durchlaufen, einige am Erweiterungs-Punkt nach der Namensauflösung. Somit bestimmt die Routenplanungs-Komponente den Weg, den eine Nachricht nimmt. Weitere Komponenten, die von der Routenplanungskomponente aufgerufen werden sind zum Beispiel die Arbeitsablaufverwaltung, der Transaktionsdienst, die Sicherheitskomponente, oder der

Standortbezogene Dienst. Bevor die Nachricht der Empfängeradresse zugestellt wird, passt die Routenplanung die Nachricht situationsspezifisch an den Empfänger an. Diese **situationsspezifische Anpassung** ist zum einen die Formatierung der Nachricht in das Protokoll des Empfängers. So wird beispielsweise eine Nachricht mit E-Mailadresse als Empfangsadresse in eine E-Mail umgewandelt oder eine Nachricht mit Telefonnummer als Empfangsadresse in eine SMS oder direkten Anruf umgewandelt. Um Nachrichten die einem speziellen Protokoll unterliegen, dem Empfänger zustellen zu können, muss im Falle einer E-Mail, SMS oder Sprachnachricht ein entsprechender Server kontaktiert werden. Die Nachricht muss somit über einen E-Mail-Server, SMS-Server oder Voice-Over-IP-Server versendet werden, der auch Protokollumsetzer genannt wird. Die Routenplanungskomponente erkennt anhand des vom Namensdienst zurückgelieferten Adresstyps das Empfangsprotokoll und liest in einer internen Konfiguration die Adresse des zuständigen Protokollumsetzers aus. Für den Fall, dass in einer Produktion keine entsprechenden Server zur Verfügung stehen, bietet die Routenplanung die Möglichkeit, die entsprechenden Server als Open Source Software zu installieren. Somit werden alle Nachrichten mit JMS-Adresse direkt zugestellt und alle anderen Nachrichten gemäß der Empfängeradresse zu einem passenden Protokollumsetzer gesendet und dort zum Empfänger wie zum Beispiel Mobiltelefon oder Posteingang weitergeleitet.

Die Besonderheit der situationsspezifischen Anpassung der Nachricht an den Empfänger ist, dass die Routenplanungs-Komponente zur Laufzeit Informationen über die verbundenen Klienten der Nachrichtenorientierten Verteilungsplattform einsammelt und in einer **Zuordnungstabelle** speichert. Dadurch kann zu jedem Zeitpunkt erkannt werden, welche Empfangsgeräte im System momentan verfügbar sind und welche Personen mit welchen Geräten angemeldet sind. In Abbildung 4-8 ist die Verknüpfung zwischen der Routenplanung, der Zuordnungstabelle und den Klienten zu erkennen.

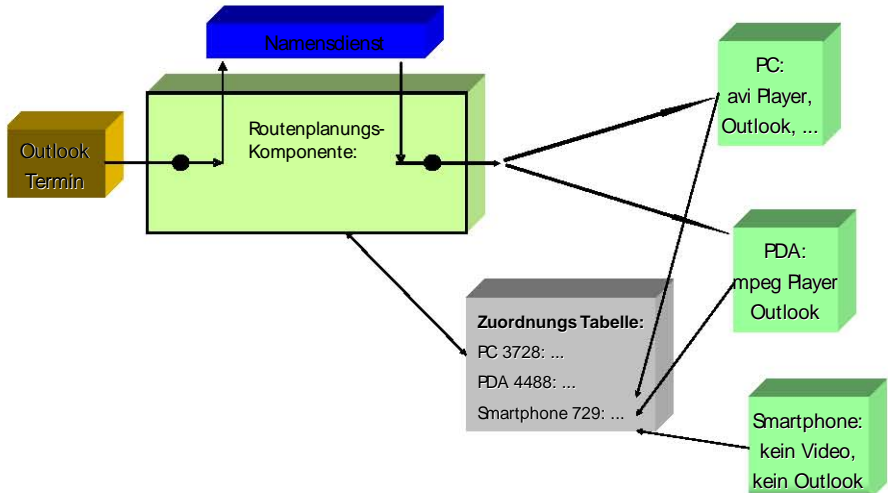


Abbildung 4-8: Situationsspezifische Routenplanung durch Zuordnungsverhandlung

Die Routenplanungskomponente wird konfiguriert, um festzulegen welche Art von Informationen von den Klienten eingesammelt werden soll. Diese Informationen sind:

- Die graphische Auflösung des Bildschirms vom Klienten in Pixel.
- Die Größe des Hauptspeichers des Klienten.
- Momentan bestehende Verbindung zur Nachrichtenorientierten Verteilungsplattform.
- Bandbreite der Verbindung vom Klienten (Netzwerk, WLAN, Bluetooth, RFID) zur Nachrichtenorientierten Verteilungsplattform.
- Das Composite Capability / Preference Profiles (CC/PP) [Klyne04] von mobilen Geräten.
- Die Verfügbarkeit eines E-Mail-Klienten.
- Die Verfügbarkeit einer Java Laufzeitumgebung zum Starten von Applets.
- Die Verfügbarkeit eines Klienten zum Betrachten von Bildern (.jpeg, .gif, .pic, ...).
- Die Verfügbarkeit eines Klienten zum Abspielen von Videodateien (.mpg, .mpeg, .avi, ...).

Durch die Zuordnungstabelle kann die Routenplanung, nachdem die Adresse des Empfängers feststeht, die Informationen des Empfangs-Klienten analysieren. Wenn nun beispielsweise vorgesehen ist, eine E-Mail an einen Empfänger zu senden, obwohl dieser momentan keinen E-Mail-Klienten besitzt, benachrichtigt die Routenplanung automatisch den Absender über die vorhandenen Empfangsprotokolle des Empfängers. Der Empfänger hat die Möglichkeit, die Routenplanung so zu konfigurieren, dass der komplette Inhalt der Nachricht in ein verfügbares Protokoll umgewandelt wird. So kann beispielsweise eine kurze E-Mail auch komplett in eine SMS umgewandelt werden. Ein häufiger Anwendungsfall bei mobilen Geräten ist die **Anpassung der Nachricht** an die graphische Auflösung. Ein von der Routenplanung gesteuerter Webserver passt hierbei die Darstellung von HTML-Seiten an die verfügbare Auflösung des Klienten an. Ebenfalls ist in der Zuordnungstabelle die Information verfügbar, welche Person mit welchen Klienten momentan mit der Nachrichtenorientierten Verteilungsplattform verbunden ist. Diese Information wird bei der Nachrichtenzustellung berücksichtigt.

Mithilfe der Zuordnungstabelle und der Funktionalität zur Anpassung der Nachricht wird die folgende Anforderung erfüllt:

- Einbeziehung aller Informationen des Empfängers für eine situationsspezifische Anpassung der Nachricht an das optimale Nachrichtenprotokoll des Empfängers und der Darstellung.

Die Routenplanung bietet den Klienten im System noch einen zusätzlichen Vorteil an. Nachrichten, die versendet werden, müssen vom Absender noch keine gültige Empfängeradresse zugewiesen bekommen haben. Nachrichten ohne gültige Empfängeradresse werden innerhalb einer Warteschlange solange zwischengespeichert, bis die Empfängeradresse zum System hinzugefügt wird. Dies ist beispielsweise von Vorteil, wenn der Empfänger noch unbestimmt ist und dieser erst später zur Laufzeit zum System definiert wird.

4.4 Entwicklung der Arbeitsablaufabstimmung

Die Arbeitsablaufabstimmungs-Komponente ermöglicht es, widersprüchliche Arbeitsabläufe von Sender und Empfänger während der Nachrichtenübermittlung aufzulösen oder zu verhandeln. Die in Kapitel 4.2 beschriebenen Profile und in Kapitel 4.1.1 erklärten Arbeitsabläufe innerhalb der Nachricht sind eine mögliche Umgebung zur Beschreibung von Arbeitsabläufen. Eine typische Ausführungsumgebung für Arbeitsabläufe ist wie in Kapitel 3.7 beschrieben die **Arbeitsablaufverwaltungs-Komponente**. In einem Enterprise Service Bus (ESB) [Chap04], der eine spezielle Form einer Nachrichtenorientierten Verteilungsplattform darstellt, ist die Arbeitsablaufverwaltung typischerweise integriert.

Die Arbeitsablaufabstimmungs-Komponente überprüft Arbeitsabläufe zur Laufzeit, ob deren Ausführung zu Widersprüchen führt. Ein Beispiel für widersprüchliche Arbeitsabläufe wurde in Kapitel 3.7.1 gegeben. Weitere Beispiele für typische Arbeitsabläufe sind:

- Sende Nachrichten zwischen 08:00 – 18:00 Uhr an die Wartungssaplikation des Wartungsarbeiters.
- Sende Nachrichten zwischen 18:00 – 08:00 Uhr als SMS an das Mobiltelefon des Wartungsarbeiters.
- Sende Nachrichten an denjenigen Wartungsarbeiter, der sich am häufigsten an der Anlage angemeldet hat.

Durch eine Konfiguration lassen sich in dieser Komponente Regeln einfügen, die widersprüchliche Arbeitsabläufe auflösen. Auch Prioritäten können für Arbeitsabläufe definiert werden. Ob es bei einer Nachrichtenübermittlung zum Einsatz der Arbeitsablaufabstimmung kommt wird von einem **Zustandsautomat** bestimmt. Der Zustandsautomat der Arbeitsablaufabstimmung ist in Abbildung 4-9 beschrieben.

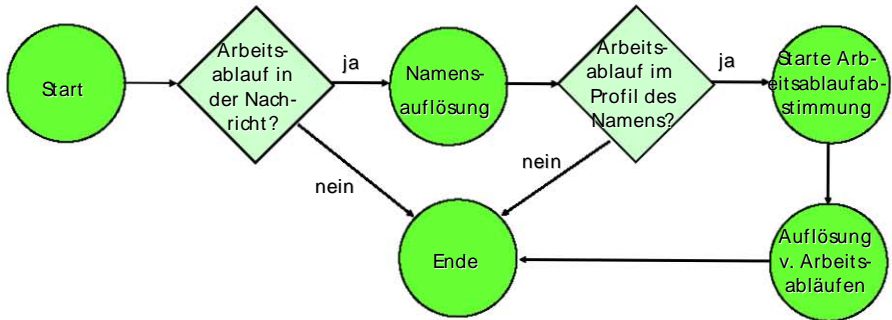


Abbildung 4-9: Zustandsautomat der Arbeitsablaufabstimmung

Für jede Nachricht, die in der Nachrichtenorientierten Verteilungsplattform eintrifft, wird der Zustandsautomat gestartet. Zuerst wird in der Routenplanung überprüft, ob der Sender der Nachricht Arbeitsabläufe mitgesendet hat. Dies wird schnell aus dem NachrichtenKopf erkannt. Wenn keine Arbeitsabläufe vom Sender zur Ausführung geplant sind, so kann es zu keinen widersprüchlichen Arbeitsabläufen kommen und der Zustandsautomat geht in den Zustand Ende über. Für den Fall, dass Arbeitsabläufe in der Nachricht definiert sind, wird in den Zustand Namensauflösung gewechselt. Wenn es im Namensraum an keinem aufzulösenden Namen ein Profil gibt, sind wiederum keine widersprüchlichen Arbeitsabläufe möglich und es wird in den Zustand Ende gewechselt. Wenn aber Arbeitsabläufe im Profil definiert sind, ist es möglich, dass widersprüchliche Arbeitsabläufe vorliegen und die Arbeitsablaufabstimmung wird gestartet. Nun werden die Arbeitsabläufe aus der Nachricht und dem Profil in die Arbeitsablaufabstimmung übernommen und dort untersucht, ob es zu Widersprüchen kommt. Wenn es zu Widersprüchen kommt, werden diese Anhand von definierten Regeln und Prioritäten aufgelöst.

Für den Fall, dass es für **widersprüchliche Arbeitsabläufe** keine definierte Regel oder Priorität gibt, geht die Arbeitsablaufabstimmungs-Komponente ähnlich wie bei Preisverhandlungen vor. Es wird eine erste **Verhandlungsrunde** gestartet, in der von jedem Klient ein Arbeitsablauf ausgeführt wird, sodass die Arbeitsabläufe sich nicht gegenseitig widersprechen. Widersprüchliche Arbeitsabläufe werden nicht berücksichtigt. Danach werden so viele weitere Verhandlungsrunden angefügt, bis alle nicht widersprüchlichen Arbeitsabläufe untergebracht sind. Die widersprüchlichen

Arbeitsabläufe gehen somit zwar verloren, aber das Gesamtsystem ist stabil und kann eine maximale Anzahl an Arbeitsabläufen ausführen. Es finden zwei verschiedene Durchläufe der Verhandlungsrunden statt. Zuerst startet Klient 1 mit einem Arbeitsablauf, danach startet in einem zweiten Durchlauf Klient 2 mit den Verhandlungsrunden. Derjenige Durchlauf, der zum besten Gesamtergebnis führt, wird übernommen. Somit wird eine Priorisierung der Arbeitsabläufe aufgrund der angeforderten Reihenfolge erreicht. Dies setzt voraus, dass die Klienten jeweils ihre wichtigsten Arbeitsabläufe an erster Stelle nennen. Es wäre auch möglich, alle Permutationen von Arbeitsabläufen durchzutesten und dabei jeweils immer einen zusätzlichen Arbeitsablauf zu entfernen. Diese Lösung hätte ein optimales Ergebnis, führt aber zu einem hohen zeitlichen und rechenintensiven Aufwand. Da die Anforderung einer schnellen Nachrichtenübermittlung besteht, wurde diese Möglichkeit ausgeschlossen.

Das beste Gesamtergebnis beruht auf der maximalen Anzahl ausführbarer Arbeitsabläufe. Abbildung 4-10 zeigt das System anhand von zwei Klienten, die jeweils verschiedene Arbeitsabläufe ausführen lassen wollen.



Abbildung 4-10: Verhandlungsrunden zur Einigung von Arbeitsabläufen

In Verhandlungsrunde 1 wird von jedem Klienten genau ein Arbeitsablauf ausgeführt und zwar Arbeitsablauf abc von Klient 1 und Arbeitsablauf xyz von Klient 2. Danach startet Verhandlungsrunde 2 mit wieder jeweils genau einem Arbeitsablauf von jedem Klienten, hier Arbeitsablauf def von Klient 1 und Arbeitsablauf uvw von Klient 2. Nun werden so viele Verhandlungsrunden durchgeführt, bis alle Arbeitsabläufe untergebracht sind. Wenn nur noch einer der beiden Klienten Arbeitsabläufe zur Ausführung offen hat, wird jeweils nur ein Arbeitsablauf pro Verhandlungsrunde zugeteilt.

Die verschiedenen Komponenten, die bei der Nachrichtenübermittlung benötigt werden um widersprüchliche Arbeitsabläufe aufzulösen sind in Abbildung 4-11 in einem Sequenzdiagramm dargestellt.

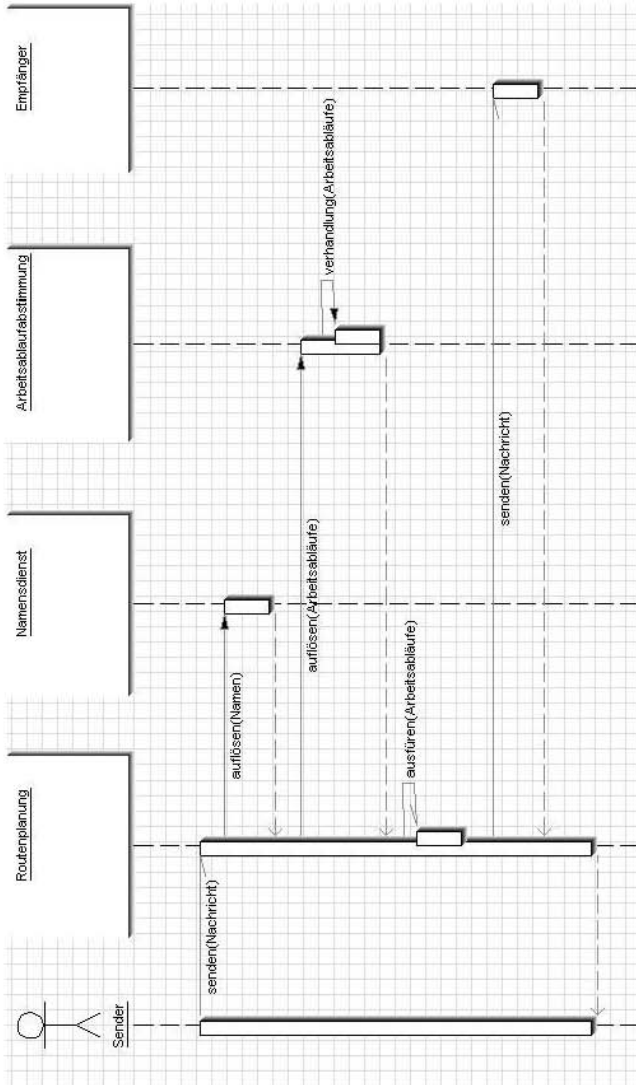


Abbildung 4-11: Sequenzdiagramm der Auflösung von Arbeitsabläufen

Die Routenplanung erhält die Nachricht eines Senders und startet den Zustandsautomat der Arbeitsablaufabstimmung. Wenn die Komponente Namensdienst aufgerufen wurde, um den Namen des Empfängers aufzulösen, werden der Routenplanung die physikalische Adresse, sowie noch auszuführende Arbeitsabläufe aus dem Profil des aufzulösenden Namens zurückgegeben. Für den Fall, dass der Zustandsautomat entscheidet, die Arbeitsablaufabstimmung aufzurufen, werden alle Arbeitsabläufe aus der Nachricht und aus dem Profil an die Arbeitsablaufabstimmung gesendet. Durch die vorhandenen Regeln und Prioritäten werden die Arbeitsabläufe aufgelöst. Für den Fall, dass keine Regeln vorhanden sind, wird die Verhandlung innerhalb der Arbeitsablaufabstimmung aktiviert. Es werden solange Arbeitsabläufe eliminiert, bis eine maximale Anzahl an Arbeitsabläufen fehlerfrei ausgeführt werden kann.

Für eine Beschreibung des Algorithmus für die Arbeitsablaufabstimmung werden folgende Definitionen benötigt:

Seien S = Sender und E = Empfänger.

Seinen x = geplanter Arbeitsablauf und y = übernommener Arbeitsablauf mit: $f(x) = y$ (Übernahme von einem geplanten Arbeitsablauf in einen auszuführenden Arbeitsablauf)

Seinen $x_{S1..n}$ = Arbeitsablauf 1..n vom Sender und $x_{E1..m}$ = Arbeitsablauf 1..m vom Empfänger.

Somit lässt sich die Formel für den Algorithmus der Arbeitsablaufabstimmung folgendermaßen beschreiben:

$\forall x_{Sk} + x_{El} = \emptyset$ mit $k = 1..n$ und $l = 1..m$:

$\forall n, m \in \mathbb{N}$: Sei $i \leq (n,m) \leq p$; $i, p \in \mathbb{N}$:

$$\sum_{j=1}^i x_{Sj} + x_{Ej} + \sum_{j=i+1}^p x_j$$

Abbildung 4-12 und Abbildung 4-13 zeigen beispielhaft Verhandlungsmöglichkeiten auf. In Abbildung 4-12 wird eine Verhandlung zwischen zwei Klienten aufgezeigt, die beide unterschiedliche Nachrichtenzustellungs-Präferenzen besitzen.

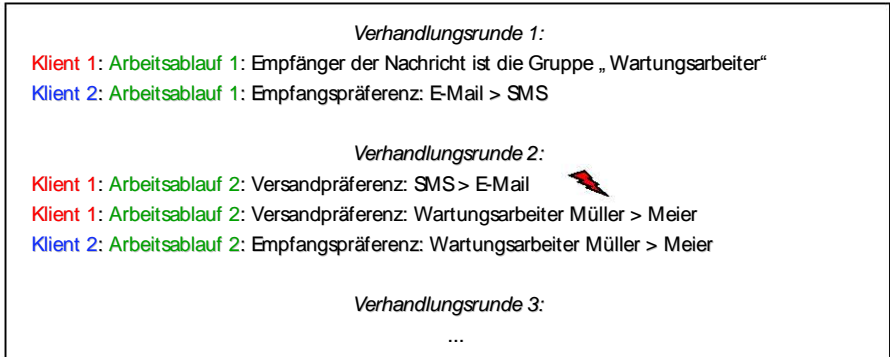


Abbildung 4-12: Nachrichtenzustellungsverhandlung

Klient 1 ist der Sender einer Nachricht und hat die Präferenzen, dass seine Nachrichten immer an die Gruppe Wartungsarbeiter zugestellt werden. Außerdem wünscht Klient 1, dass seine Nachrichten zuerst per SMS an den Wartungsarbeiter gesendet werden. Dieser Arbeitsablauf kann aber nicht ausgeführt werden und wird somit verworfen, da Klient 2 schon in Verhandlungsrunde 1 festgelegt hat, dass Nachrichten an die Gruppe Wartungsarbeiter immer zuerst als E-Mail und danach als SMS zugestellt werden sollen. Arbeitsablauf 2 von Klient 2 sagt aus, dass Nachrichten zuerst an Wartungsarbeiter Müller vor Meier gesendet werden stimmt mit dem neuen Arbeitsablauf 2 von Klient 1 überein und wird somit ausgeführt.

Nachdem der in Abbildung 4-12 gezeigte Verhandlungsdurchlauf abgeschlossen ist, findet ein zweiter Verhandlungsdurchlauf statt, beginnend mit einem Arbeitsablauf von Klient 2. Abbildung 4-13 zeigt diesen zweiten Verhandlungsdurchlauf.

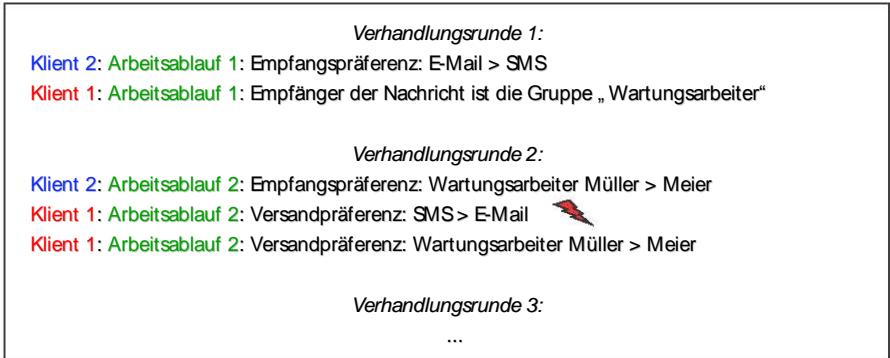


Abbildung 4-13: Verworfenen Verhandlungsdurchlauf

Da dieser Verhandlungsdurchlauf genauso viele Arbeitsabläufe wie Verhandlungsdurchlauf 1 durchführen kann, wird er verworfen. Nur bei einer Anzahl von Arbeitsabläufen höher als im ersten Verhandlungsdurchlauf wird der zweite Verhandlungsdurchlauf übernommen.

In Abbildung 4-14 ist die Auflösung und Verhandlung von Arbeitsabläufen durch die Arbeitsablaufabstimmung dargestellt.

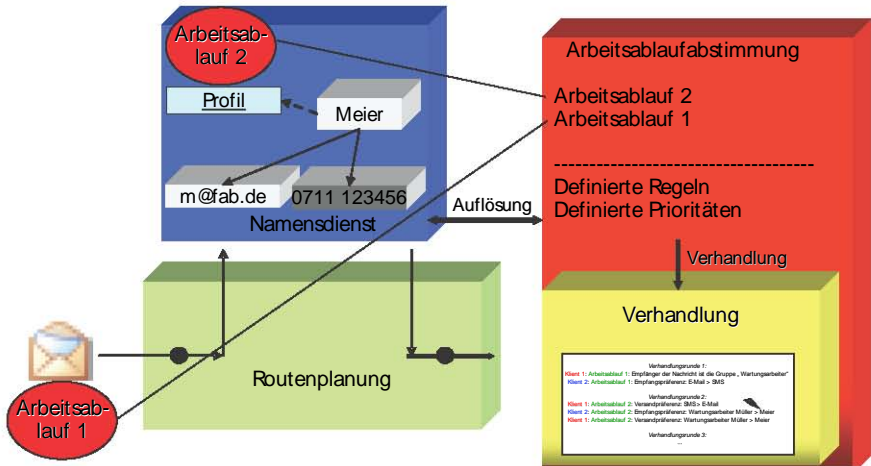


Abbildung 4-14: Auflösung und Verhandlung der Arbeitsablaufabstimmung

Die verschiedenen Arbeitsabläufe aus der Nachricht und aus dem Profil werden in der Arbeitsablaufabstimmung gespeichert. Innerhalb einer XML-Datei sind Regeln und Prioritäten definiert, die von der Arbeitsablaufabstimmung benutzt werden, um Arbeitsabläufe aufzulösen. Wenn es dennoch zu Widersprüchen bei der Ausführung der Arbeitsabläufe kommen würde, werden innerhalb einer Verhandlung die Störenden **Arbeitsabläufe eliminiert**.

Die Arbeitsablaufabstimmung wurde als leichtgewichtige und schnelle regelbasierte Komponente realisiert. Die Arbeitsabläufe werden durch Regeln behandelt und priorisiert. Somit erfüllt die Arbeitsablaufabstimmung die Anforderung einer:

- Auflösung/Verhandlung von konkurrierenden Arbeitsabläufen.

Die Anforderung

- Situationsspezifische Abstimmung von Arbeitsabläufen

wird dadurch erfüllt, dass während der Verhandlung von Arbeitsabläufen zur Laufzeit immer das System überprüft wird, welche Klienten mit welchen Eigenschaften momentan verfügbar sind.

4.5 Integration der Komponenten in das Gesamtsystem

Die Komponenten aus Kapitel 4.2, Kapitel 4.3, und Kapitel 4.4 werden zusammen in die Nachrichtenorientierte Verteilungsplattform integriert um die in Kapitel 4.1 abgeleiteten Anforderungen des Gesamtsystems zu erfüllen. Abbildung 4-15 zeigt eine Nachrichtenorientierte Verteilungsplattform mit den entwickelten Komponenten Namensdienst und Routenplanung und der neuen Komponente Arbeitsablaufabstimmung.

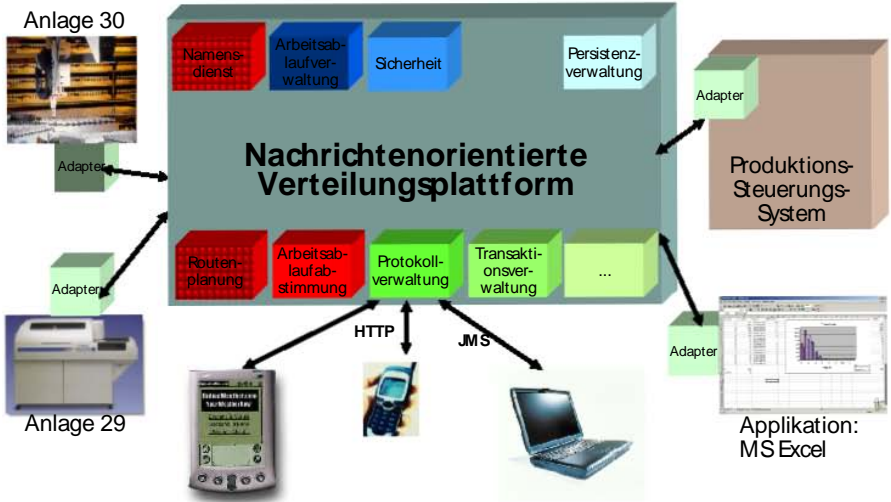


Abbildung 4-15: Nachrichtenorientierte Verteilungsplattform mit Innovativen Komponenten

Abbildung 3-8 aus dem Stand der Technik wurde modifiziert, um die innovativen Komponenten, die in Kapitel 4.2, Kapitel 4.3, und Kapitel 4.4 entwickelt wurden, aufzuzeigen.

Die Abhängigkeiten und Beziehungen der verschiedenen Komponenten innerhalb der Nachrichtenorientierten Verteilungsplattform sind in Abbildung 4-16 dargestellt.

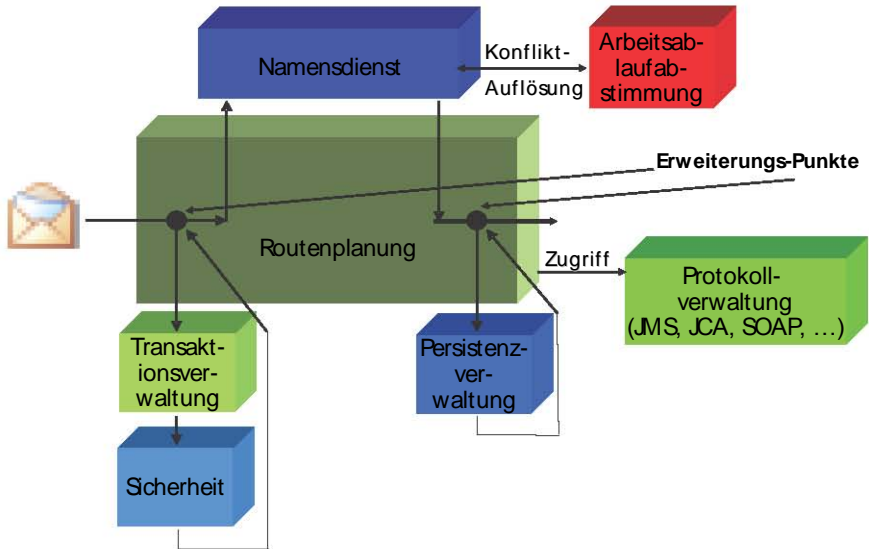


Abbildung 4-16: Beziehungen zwischen den Komponenten der Nachrichtenorientierten Verteilungsplattform

Die zentrale Komponente der Nachrichtenorientierten Verteilungsplattform ist die Routenplanung. Sie nimmt eingehende Nachrichten entgegen und verwaltet sie. Sie bestimmt in welcher Reihenfolge die anderen Komponenten die Nachricht erhalten und ruft den Namensdienst zur Namensauflösung auf. Nur durch die Zusammenarbeit von Routenplanung und Namensdienst wird erreicht, dass die Routenplanung aufgrund des im Namensgraphen gespeicherten Adresstyps die Konvertierung der Nachricht in das Empfangsprotokoll vornehmen kann. Da die Arbeitsablaufabstimmung die Ausführung von Arbeitsabläufen in Profilen im Namensdienst und in der Arbeitsablaufverwaltungs-Komponente überwacht und koordiniert, müssen die Komponenten Namensdienst und Arbeitsablaufabstimmung zusammen integriert werden, um zu gewährleisten, dass widersprüchliche Arbeitsabläufe aufgelöst werden. Durch die Datenbankverbindungs-Komponente wird der Namensbaum persistiert. Die Routenplanung hat direkten Zugriff zur Protokollverwaltung, um die Nachricht in das Protokoll des Empfängers zu wandeln.

An den Erweiterungspunkten wird die Nachricht an weitere Komponenten der Nachrichtenorientierten Verteilungsplattform weitergeleitet. Zusätzlich zu den

Komponenten Transaktionsdienst, Sicherheit und Arbeitsablaufverwaltung können eigene Komponenten der Nachrichtenorientierten Verteilungsplattform zugefügt werden, denen die Nachricht zugestellt wird. Nachdem die Nachricht allen Komponenten am Erweiterungs-Punkt nach der Namensauflösung zugestellt wurde, verlässt sie die Nachrichtenorientierte Verteilungsplattform und wird dem Empfänger oder einem Protokollumwandler zugestellt.

5 Realisierung und Anwendung

Die Komponenten dieser Arbeit sind in der Programmiersprache Java mit dem Standard **J2EE 1.4** implementiert. Für den Namensdienst, die Routenplanung und die Arbeitsablaufabstimmung wurde jeweils eine neue Komponente, basierend auf der Konzeption, in Java implementiert.

Die Integration der Komponenten erfolgt dadurch, dass die Komponenten als **Managed Bean** (MBean) in Java implementiert sind und in einem **Anwendungs-Server** (engl. Application-Server) [Ihns03] eingesetzt werden. Dadurch ist es möglich die Komponenten zur Laufzeit der Nachrichtenorientierten Verteilungsplattform hinzuzufügen oder zu entfernen. Mit dieser Technik kann die Nachrichtenorientierte Verteilungsplattform in der Produktion im 24/7 Betrieb weiterlaufen, während einzelne Komponenten zur Laufzeit ausgetauscht werden können. In dieser Realisierung handelt es sich um den Open-Source Anwendungs-Server JBoss [JBoss07]. Der Anwendungs-Server verwaltet alle Komponenten als MBean und stellt außerdem einen Kommunikations-Bus dar. Für die Nachrichtenübermittlung auf den unteren OSI-Schichten wird JBossMQ [JBossMQ07] verwendet. Eine verteilte Persistenz wird in JBoss durch JavaGroups [JGroups07] erreicht. Bei der Entwicklung der Komponenten dieser Arbeit wurde auf die standardisierte Verwendung von **JMS** geachtet. Bestehende Nachrichtenorientierte Verteilungsplattformen auf Java Basis können somit die Komponenten direkt integrieren. Nur wenige Konfigurations-Dateien müssen angepasst werden, damit die Komponenten innerhalb verschiedener Nachrichtenorientierter Verteilungsplattformen verwendbar sind. In Abbildung 5-1 ist der Anwendungs-Server JBoss mit der Nachrichtenorientierten Verteilungsplattform als MBean dargestellt.

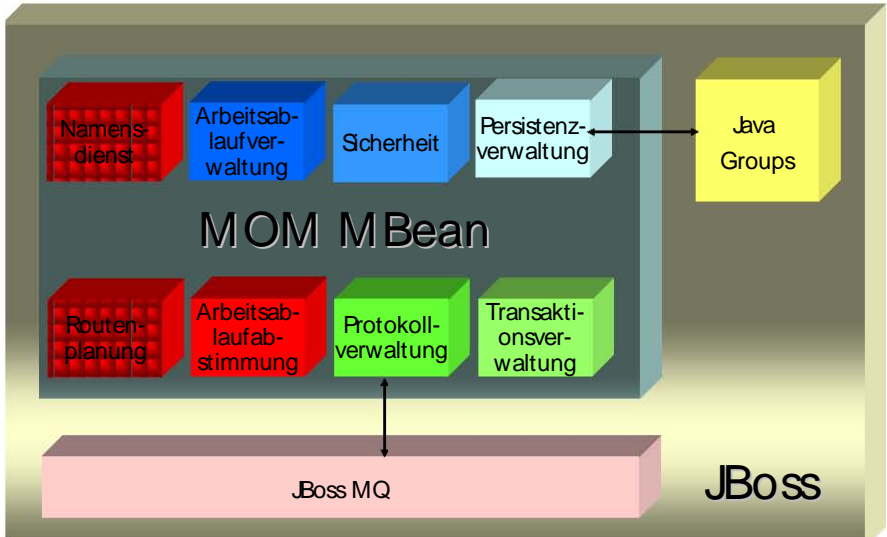


Abbildung 5-1: Nachrichtenorientierte Verteilungsplattform als MBean innerhalb des Anwendungs-Server

In den folgenden Unterkapiteln wird auf die Realisierung der einzelnen Komponenten eingegangen und zusammenfassend die integrierte Lösung betrachtet.

5.1 Namensdienst

Der Namensdienst bietet eine grafische Oberfläche, in der Administratoren und Wartungsarbeiter den Namensraum verwalten können. Abbildung 5-2 zeigt die grafische Oberfläche zum Anlegen von Namen und Adressen und deren Verknüpfung untereinander.

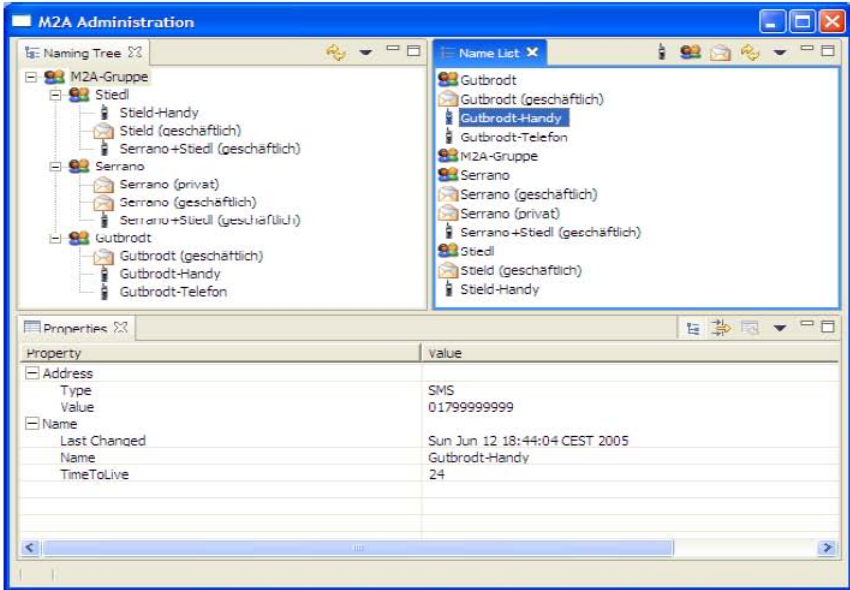


Abbildung 5-2: Grafische Oberfläche für den Namensdienst

Die grafische Oberfläche für den Namensdienst ist einfach gehalten, damit sie in der laufenden Produktion ohne lange Einarbeitungszeit von Wartungsarbeitern bedient werden kann. Eine Unterteilung in drei Fenster wurde vorgenommen. Innerhalb des Fensters „Naming Tree“ wird der Namensraum dargestellt. Jedem Namen und jeder Adresse wird ein Symbol für Person, Gruppe, Applikation, Anlage, Mobiles Gerät, Email, Telefon, zugeordnet. Im Fenster „Name List“ werden alle verfügbaren Namen und Adressen alphabetisch geordnet ohne Hierarchie dargestellt. Das Fenster „Properties“ zeigt die Eigenschaften des momentan ausgewählten Objekts. Die Eigenschaften für eine Adresse sind der Typ und der Wert. Für einen Namen definieren die Eigenschaften, wann das Objekt zuletzt geändert wurde, der Name und die Zeit, wie lange das Objekt mindestens noch verfügbar ist.

Die grafische Oberfläche erlaubt, Namen und Adressen zu erstellen, zu lesen, zu ändern und zu löschen. Zwischen den Namen und Adressen können Abhängigkeiten erstellt und gelöscht werden.

Profile können ebenfalls zu jedem Namen erstellt und gelöscht werden. Dabei können innerhalb eines Profils Arbeitsabläufe erstellt, gelesen, geändert und gelöscht werden.

Um eine schnelle persistente Namensauflösung zu gewährleisten, wurde das Modell des Namensdienstes mit **Enterprise Java Beans (EJB) 3.0** umgesetzt. EJB bieten die flexibelste Architektur für Verteilungsplattformen an [Cecchet03]. Die Speicherung der Objekte im Namensgraphen kann über eine beliebige **Java Data Base Connector (JDBC)**-kompatible Datenbank erfolgen.

Besonderheit innerhalb der Komponente Namensdienst sind die selbst entworfenen Objekttypen für die unterschiedlichen Adressen. Damit wird anhand des Objekttyps erkannt, welches Nachrichtenprotokoll der Empfänger besitzt.

Als Datenstruktur A eines gerichteten azyklischen Graphen wird eine doppelt verkettete Liste verwendet um zusätzlich eine Liste der Vorgänger zu verwalten. Sei $G = (V, E)$ ein Graph und $x, y \in E$ mit $A(x) = \{y \in V : x, y \in E\}$.

5.2 Routenplanung

Die Komponente Routenplanung bietet keine Benutzeroberfläche an, da sie eine rein funktionelle Komponente für die Nachrichtenübermittlung darstellt. Bei jeder ankommenden Nachricht wird immer die Schnittstelle „**public Set resolveName (Name name)**“ zur Komponente Namensdienst aufgerufen. Dahinter verbirgt sich die Namensauflösung mit anschließender Ausführung von vorhandenen Arbeitsabläufen am Profil des aufzulösenden Namens. Der Namensdienst bekommt nicht die komplette Nachricht zugestellt, sondern übernimmt die Namensauflösung für die Routenplanung. Der Routenplanung wird eine Liste der physikalischen Empfängeradressen zurückgegeben.

Jeweils vor oder nach der Namensauflösung wird an den Erweiterungs-Punkten die Nachricht durch die Funktion „**public synchronized void forwardMessage (Message message)**“ weitergeleitet. Durch die Funktionen „**public void setRoute (Route route)**“ und „**protected void**

`addAddressesToRoute(Route route, Name name)`“ kann ein neuer Weg innerhalb der Komponenten der Nachrichtenorientierten Verteilungsplattform bestimmt werden. Außerdem kann durch diese zwei Funktionen auch ein Protokollumsetzer als Ziel für die Nachricht definiert werden.

5.3 *Arbeitsablaufabstimmung*

Die Komponente Arbeitsablaufabstimmung wird während der Namensauflösung aktiviert, falls die Nachricht vom Sender Arbeitsabläufe enthält und gleichzeitig Arbeitsabläufe am Profil des aufzulösenden Namens auszuführen sind.

Regeln für die Auflösung und Verhandlung von widersprüchlichen Arbeitsabläufen werden innerhalb einer **XML Konfigurationsdatei** angegeben. Die XML Datei mit einfachen Regeln und Prioritäten ist im Folgenden dargestellt:

```
<?xml version="1.0" encoding="UTF-8" ?>
<arbeitsablaufabstimmung>
  <rules>
    <rule>if message.sender="polish_equipment_12" then
      message.receiver="Michael Meier"</rule>
    <rule>if message.sender="polish_equipment_15" then
      message.receiver="Jackob Mueller"</rule>
  </rules>
  <priorities>
    <priority>if message.name="alert" then priority="highest"</priority>
  </priorities>
</arbeitsablaufabstimmung>
```

Die erste Regel sagt aus, dass jede Nachricht von der Anlage `polish_equipment_12` immer direkt an den Wartungsarbeiter Michael Meier zugestellt werden soll. In der zweiten Regel ist definiert, dass jede Nachricht von der Anlage `polish_equipment_15` immer direkt an den Wartungsarbeiter Jackob Mueller zugestellt werden soll. Als Priorität ist festgelegt, dass alle Nachrichten mit dem Namen Alarm, die höchste Priorität bei der Nachrichtenzustellung haben.

5.4 Anwendung der integrierten Nachrichtenorientierten Verteilungsplattform

Die in Kapitel 4.2, 4.3 und 4.4 dargestellten Komponenten stellen die Basis für eine Nachrichtenorientierte Verteilungsplattform dar. Zusammen mit weiteren typischen Komponenten einer Nachrichtenorientierten Verteilungsplattform wie in Abbildung 4-15 ersichtlich, wurden die Komponenten innerhalb von **zwei Forschungsprojekten** bei 8 verschiedenen Endanwendern aus der Industrie eingesetzt. Diese weitere Komponenten, verschiedene Applikationen für die Produktion und ein universeller Adapter sind von Projektpartnern entwickelt worden und werden deshalb in dieser Arbeit nicht näher vorgestellt.

Innerhalb des InnoNet Forschungsprojektes „**Message to Anywhere (M2A)**“ [Knoll04c, Knoll06a, Knoll03], gefördert vom VDI/VDE-IT wurde eine Nachrichtenorientierte Verteilungsplattform für die Halbleiterindustrie [Knoll04b] und Laborautomatisierung [Knoll04a] entwickelt. In diesem Projekt wurde die Implementierung zuerst innerhalb von zwei Umgebungen getestet, die die Umgebung einer Halbleiterfabrik simulierten. Zum einen in einem bereits existierenden Testbett der Firma Systema GmbH, die Industriepartner in diesem Forschungsprojekt war und die Ihren Kunden Integrationsdienstleistungen für Nachrichtenorientierte Verteilungsplattformen anbietet. Zum anderen innerhalb einer Testumgebung, die von den Forschungspartnern Institut für Automatisierungs- und Softwaretechnik (IAS) der Universität Stuttgart und Fraunhofer IPA in Rechnerpools an der Universität Stuttgart aufgebaut wurde. Mehrere hundert Klienten wurden erstellt, die gleichzeitig Nachrichten senden, um die Geschwindigkeit und Stabilität der Nachrichtenorientierten Verteilungsplattform zu testen. In Kapitel 1 wird die Konfiguration dieser Testumgebung detailliert beschrieben. Nachdem durch die Testumgebungen die integrierte Nachrichtenorientierte Verteilungsplattform validiert und verifiziert wurde, wurde sie prototypisch bei zwei Endanwendern installiert. Einer der Endanwender innerhalb des Forschungsprojektes „Message to Anywhere“, die acp GmbH, hat das Halbleitermessgerät ParticleScan mit der Nachrichtenorientierten Verteilungsplattform verbunden. Der zweite Endanwender, die Qualitytype AG, hat die Nachrichtenorientierte Verteilungsplattform innerhalb der Bioinformatik eingesetzt. In diesem Szenario wurde die Nachrichtenorientierte

Verteilungsplattform mit der Qualitätssicherungssoftware Qualiproof verbunden. Verschiedene Nutzer von Qualiproof, wie Labore, Einzel- und Großhändler in der Lebensmittelproduktion, Tierärzte und Bauernhöfe haben durch die Nachrichtenorientierte Verteilungsplattform einen schnelleren Zugriff auf Daten erhalten. Wegen einer hohen Anzahl von Datenbankabfragen war das System ohne die Nachrichtenorientierte Verteilungsplattform an die Leistungsgrenzen gestoßen und konnte einen höheren Durchsatz an Nachrichten nicht gewährleisten.

Das zweite Forschungsprojekt „**Application Bus for Interoperability In enlarged Europe SMEs (Abilities)**“ [Knoll06b, Tolle07, Knoll07], in dem die Komponenten wiederholt zum Einsatz kommen, ist von der Europäischen Union gefördert. Hier werden das Konzept der Nachrichtenübermittlung und die Komponente Namensdienst wieder verwendet. Die Routenplanung war durch den zugrunde liegenden ESB Servicemix [Ser07] vordefiniert und es wurden zusätzlich einige nützliche Funktionalitäten von dieser Komponente dieser Arbeit übernommen. Für die Arbeitsablaufabstimmung wurde eine neue agentenbasierte Komponente entwickelt, die nicht nur Arbeitsabläufe der Nachrichtenzustellung, sondern alle Arten von Arbeitsabläufen auflöst. Der Einsatz dieser Komponenten findet auf den Servern der Unternehmen Cassovia (Slowakei), Krasme (Litauen), Innova (Türkei), Filbac (Rumänien) und Matisz (Ungarn) statt. Verschiedene Geschäftspartner aus Branchen der Hochautomatisierung, des Einzelhandels, der Agrarwirtschaft, der Holzindustrie und der Tourismusindustrie greifen auf diese Server zu und verbinden sich mithilfe der Nachrichtenorientierten Verteilungsplattform mit einem E-Business Portal. Die Nachrichtenorientierte Verteilungsplattform wird hier als Grundlage für E-Business und der Geschäftsprozess-Verwaltung eingesetzt.

6 Verifikation und Bewertung

In den folgenden Unterkapiteln findet eine Untersuchung statt, ob die in Kapitel 4.1 abgeleiteten Anforderungen durch die Komponenten dieser Arbeit erfüllt werden.

6.1 Punkt-zu-Punkt Nachrichtenübermittlung mit höherem Durchsatz als bei bestehenden Systemen

Die Nachrichtenübermittlungsgeschwindigkeit der Nachrichtenorientierten Verteilungsplattform mit den eingesetzten Komponenten aus Kapitel 4.2, 4.3 und 4.4 liegt bei bis zu 8000 Nachrichten / Sekunde. Im Durchschnitt wird eine Nachrichtenübermittlungsgeschwindigkeit von 6000 Nachrichten / Sekunde erzielt. Da in der Produktion wenige Tausend Nachrichten pro Sekunde momentan noch ausreichend sind, werden die Laufzeiterwartungen erfüllt. Um den Geschwindigkeitsgewinn zu detaillieren, wurden Laufzeittests der Komponenten dieser Arbeit mit den am Markt erhältlichen schnellsten Kommerziellen und schnellsten Open-Source Nachrichtenorientierten Verteilungsplattform durchgeführt. Die folgenden Ergebnisse der Laufzeittests stammen aus der Installation der Nachrichtenorientierten Verteilungsplattform am IAS der Universität Stuttgart. Die veröffentlichten Ergebnisse [Stiedl04] und die Konfiguration des **Testaufbaus** kann detailliert nachvollzogen werden [Stiedl04, Knoll06a]. Es handelte sich um Computer mit 1-2 GHz, 256 MB oder mehr RAM und einem Netzwerk mit 100 MBit. Als Betriebssystem war Windows XP installiert. Die Klienten wurden mit Eclipse entwickelt und Java 1.4.2 diente als Ausführungsumgebung für die Klienten. In Abbildung 6-1 ist der Testaufbau skizziert.

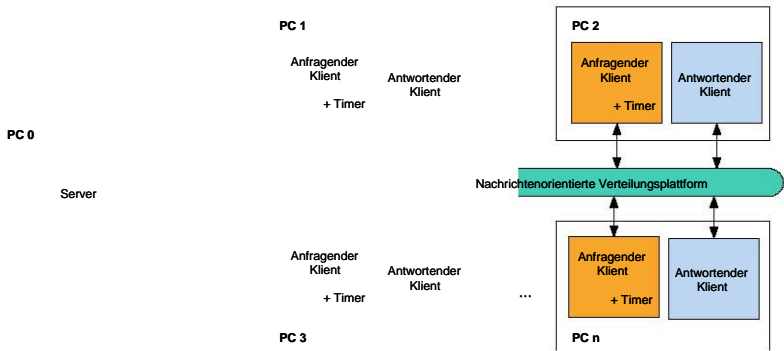


Abbildung 6-1: Testaufbau der Nachrichtenorientierten Verteilungsplattform am IAS

Der Server hat die Komponenten der Nachrichtenorientierten Verteilungsplattform installiert und führt diese aus. Die Klienten verteilen sich auf die übrigen n Computer. Innerhalb der Testphase wurde die Anzahl der Klientenpaare variiert mit den Werten 1, 10, 50 und 100. So lange auf einem Computer nicht mehr als 20 Klientenpaare anzufinden waren, bestehend aus Anfragendem und Antwortendem Klient, gab es innerhalb dieser Versuchsreihe keine Überlastung von CPU, RAM oder Netzwerkbandbreite. Jedes der Klientenpaare hat im Testversuch jeweils 200, 2.000 oder 20.000 Nachrichten versandt, im Langzeittest sogar 2.000.000 Nachrichten.

Im Folgenden wird der Ablauf des **Laufzeittests** beschrieben. Der Anfragende Klient versendet zuerst eine Nachricht an den Antwortenden Klient. Dies wird dadurch erreicht, dass eine Verbindung mit der Nachrichtenorientierten Verteilungsplattform aufgebaut wird und die Nachricht an den Server gesendet wird. Der Server leitet die Nachricht dann direkt an den Antwortenden Klient weiter. Sobald der Antwortende Klient die Nachricht erhalten hat, wird eine Nachricht als Antwort zurück an den Anfragenden Klient gesendet. Wenn der Anfragende Klient die Nachricht erhalten hat, wird diese Prozedur nun 100, 1.000, 10.000 oder 1.000.000 Mal wiederholt. Der Anfragende Klient besitzt einen Zeitmesser (eng. Timer), der die Zeit jeder Nachricht in Millisekunden, sowie die gesamte Zeit des Laufzeittests stoppt. Die Nachrichtengröße wurde innerhalb der Tests variiert. Da hier aber kaum Laufzeitveränderungen stattfanden, wurden die Testergebnisse mit 10 kByte Nachrichteninhalt veröffentlicht.

In Abbildung 6-2 ist die Architektur der Laufzeittests des Forschungsprojektes M2A mit den Komponenten der Nachrichtenorientierten Verteilungsplattform beschrieben.

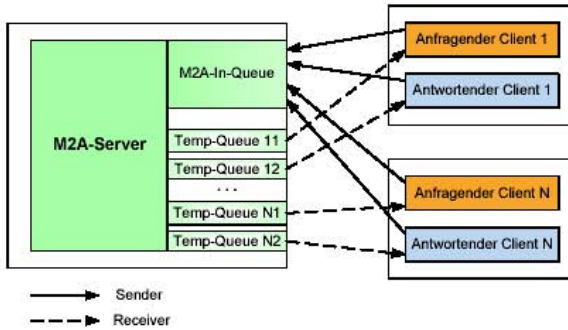


Abbildung 6-2: Architektur der M2A Tests

Ein Anfragender Klient sendet eine Nachricht immer an dieselbe Eingangs-Warteschlange (M2A-In-Queue). Nachdem die Nachricht durch die Nachrichtenorientierte Vermittlungsplattform weitergeleitet wurde, wird sie an die Warteschlange des Antwortenden Klienten gesendet. Der Antwortende Klient kann nun direkt auf die Nachricht zugreifen, und sendet sie gleich wieder zurück über die Eingangs-Warteschlange zum anfragenden Klient. In Tabelle 3 sind die Testdurchläufe mit den Gesamtzeiten, die im Durchschnitt erzielt wurden abgebildet.

Testdurchlauf	Testbeschreibung	Gesamtzeit in ms
1	1 Klientenpaar mit 100 Nachrichten pro Klient	2.824
2	1 Klientenpaar mit 1000 Nachrichten pro Klient	18.883
3	10 Klientenpaare mit 100 Nachrichten pro Klient	24.526
4	10 Klientenpaare mit 1000 Nachrichten pro Klient	158.812
5	50 Klientenpaare mit 100 Nachrichten pro Klient	123.011
6	50 Klientenpaare mit 1000 Nachrichten pro Klient	912.362
7	50 Klientenpaare mit 1000 Nachrichten pro Klient	247.948
8	100 Klientenpaare mit 1000 Nachrichten pro Klient	2.522.587
9	1 Klientenpaar mit 10000 Nachrichten pro Klient	184.826
10	10 Klientenpaare mit 10000 Nachrichten pro Klient	1.746.401
11	100 Klientenpaare mit 10000 Nachrichten pro Klient	18.368.323

Tabelle 3: Ergebnisse der Testdurchläufe

Interessant ist für die oben aufgeführten Ergebnisse die Verteilung der Zustellzeit für jede einzelne Nachricht. In Abbildung 6-3 bis Abbildung 6-13 ist für jeden Testversuch die Zustellzeit vom Absenden der Nachricht des anfragenden Klienten bis zum Empfangen der Antwort auf diese Nachricht vom Anfragenden Klient festgehalten. Die x-Achse zeigt die Zeit pro Nachricht in Millisekunden und die y-Achse zeigt die Anzahl der Nachrichten. Somit beinhaltet ein Punkt in einer der folgenden Abbildungen jeweils zwei Nachrichten, die Nachricht vom anfragenden Klient und die Antwort vom antwortenden Klient. Bei den Tests in Abbildung 6-3 und Abbildung 6-4 mit jeweils nur einem Anfragenden und Antwortenden Klient sieht man, dass eine Nachricht bei 100 Nachrichten pro Klient durchschnittlich 28,2 ms benötigt. Wenn 1000 Nachrichten pro Klient versendet werden, benötigt jede Nachricht nur noch 18,9 ms. Die Achsenbeschriftungen sind in folgenden Diagrammen für die senkrechte Achse „Übertragungszeit pro Nachricht in ms“ und für die wagerechte Achse „Anzahl der Nachrichten“

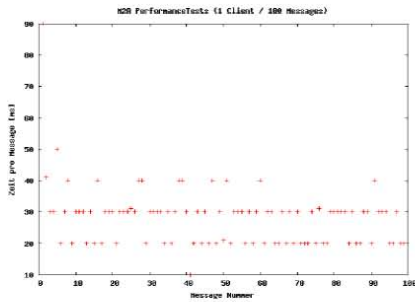


Abbildung 6-3: Test mit 1 Klientenpaar und 100 Nachrichten / Klient

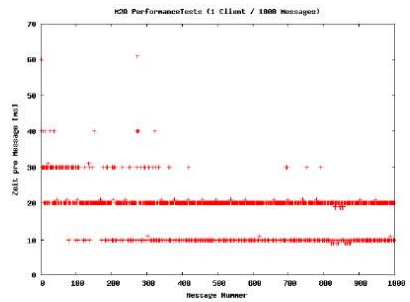


Abbildung 6-4: Test mit 1 Klientenpaar und 1000 Nachrichten / Klient

Durch die Erhöhung der Klientenpaare von 1 auf 10 findet auch eine Erhöhung der durchschnittlichen Nachrichtenzustellzeit statt. In Abbildung 6-5 benötigt eine Nachricht bei 10 Klientenpaaren mit je 100 Nachrichten pro Klient im System durchschnittlich 24,5 ms. Bei 10 Klientenpaaren mit je 1000 Nachrichten pro Klient verkürzt sich die durchschnittliche Zustellzeit bereits auf 15,9 ms, wie in Abbildung 6-6 zu erkennen ist.

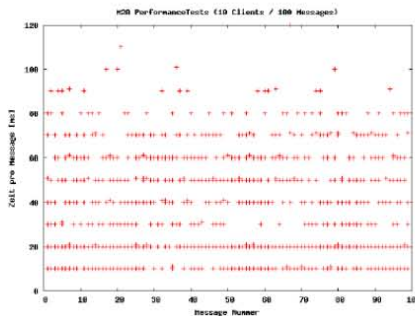


Abbildung 6-5: Test mit 10 Klientenpaaren und 100 Nachrichten / Klient

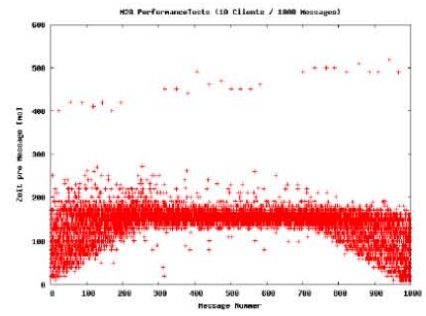


Abbildung 6-6: Test mit 10 Klientenpaaren und 1000 Nachrichten / Klient

In Abbildung 6-7 und Abbildung 6-8 wurde die Anzahl der Klientenpaare auf 50 erhöht. Die Zustellzeit einer Nachricht bei 100 Nachrichten pro Klient beträgt 24,6 ms. Bei 1000 Nachrichten pro Klient verkürzt sich die Zustellzeit auf 18,2 ms.

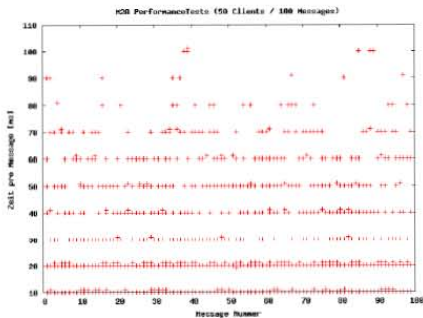


Abbildung 6-7: Test mit 50 Klientenpaaren und 100 Nachrichten / Klient

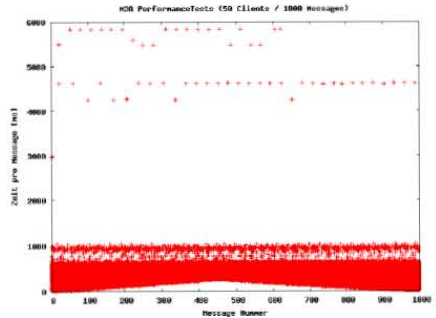


Abbildung 6-8: Test mit 50 Klientenpaaren und 1000 Nachrichten / Klient

In Abbildung 6-9 und Abbildung 6-10 wurden 100 Klientenpaare gleichzeitig getestet. Die durchschnittliche Zustellzeit einer Nachricht benötigt bei 100 Nachrichten pro Klient 24,8 ms. Bei einer Erhöhung auf 1000 Nachrichten pro Klient verlängert sich die Zustellzeit auf durchschnittlich 25,2 ms.

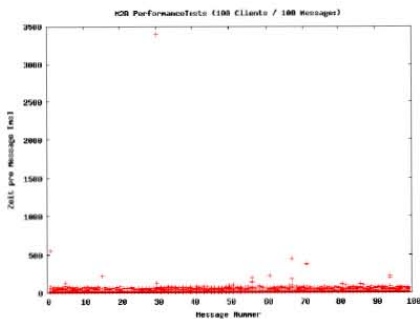


Abbildung 6-9: Test mit 100 Klientenpaaren und 100 Nachrichten / Klient

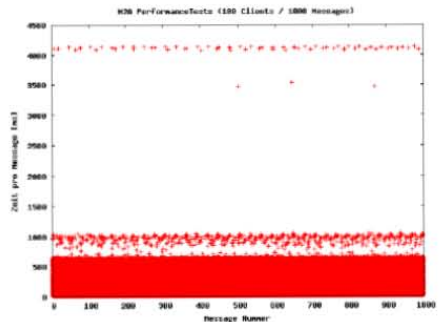


Abbildung 6-10: Test mit 100 Klientenpaaren und 1000 Nachrichten / Klient

In Abbildung 6-11, Abbildung 6-12 und Abbildung 6-13 ist die Anzahl der Nachrichten pro Klient auf 10.000 erhöht worden.

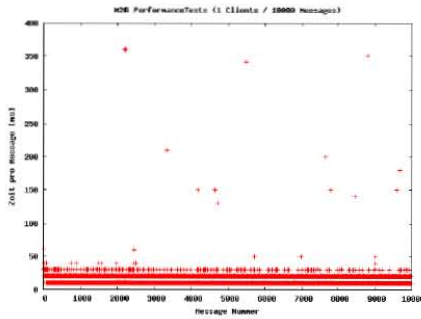


Abbildung 6-11: Test mit 1 Klientenpaar und 10000 Nachrichten / Klient

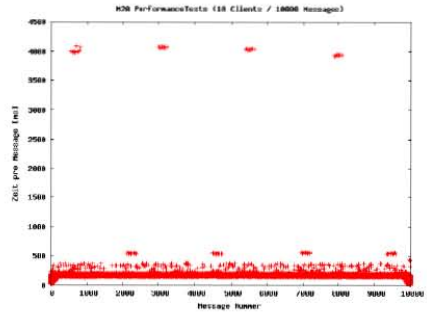


Abbildung 6-12: Test mit 10 Klientenpaaren und 10000 Nachrichten / Klient

Mit einem Klientenpaar beträgt die durchschnittliche Zustellzeit 18,5 ms. Mit einer Erhöhung auf 10 Klientenpaare, verkürzt sich die durchschnittliche Zustellzeit auf 17,5 ms. Bei 100 Klientenpaaren erhöht sich wiederum die durchschnittliche Zustellzeit auf 18,4 ms.

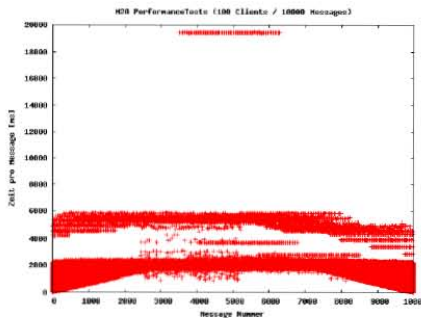


Abbildung 6-13: Test mit 100 Klientenpaaren und 10000 Nachrichten / Klient

Besonders auffällig ist, dass sich der Großteil der Nachrichten auf derselben Zustellzeit einpendelt, es aber regelmäßig einen kleinen Anteil an Ausnahmen gibt, die eine wesentlich längere Zustellzeit benötigen. Dies liegt an dem auf ISO-OSI Schicht 3-4 liegenden Protokoll, das in der getesteten Version noch Laufzeitanomalien aufwies.

Während der Testversuche wurden die CPU-Auslastung, die RAM-Auslastung und die Netzwerkbelastung überwacht. In Abbildung 6-14 ist der **CPU-Auslastungs-Verlauf** für die Langzeittests mit 1.000.000 Nachrichten pro Klient zu sehen.

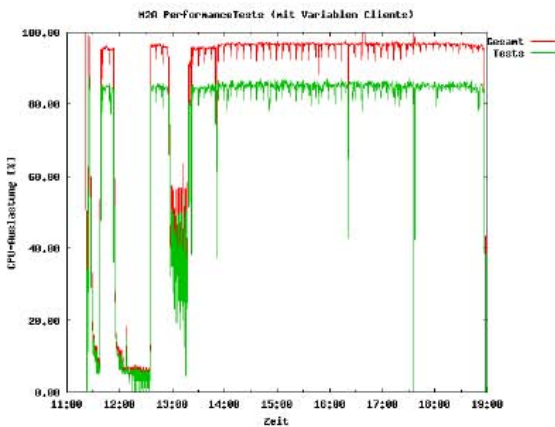


Abbildung 6-14: CPU Auslastung des Servers

Die Auslastung der CPU des Servers war bei den Testversuchen bei ca. 80%. Eine höhere Last darüber hinaus hätte eine Erhöhung der Zustellzeit zur Folge. Deswegen wurde spätestens ab einer Auslastung von 85% ein neuer Server repliziert, der dann innerhalb des Verteilten Systems die zusätzliche Arbeit übernahm.

Gegenüber der CPU-Auslastung, wurde bei einem 100 MBit Netzwerk die Bandbreite kaum ausgenutzt. Selbst bei den Langzeittests wurde nie mehr als 6% der Bandbreite beansprucht. Abbildung 6-15 zeigt die **Auslastung der Bandbreite** des Servers während des Langzeittests.

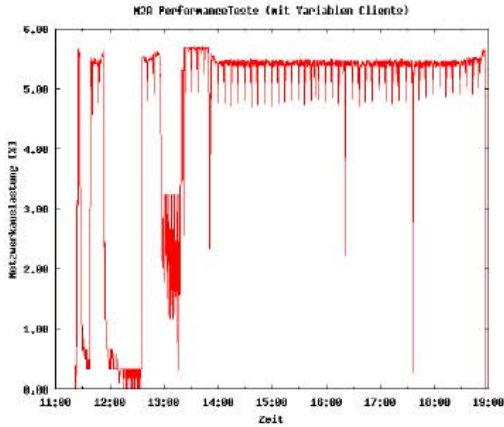


Abbildung 6-15: Netzwerkauslastung des Servers

Dieser Testversuch wurde mit verschiedenen Nachrichtenorientierten Verteilungsplattformen durchgeführt. In [Stiedl04, Knoll06a] sind jeweils die schnellste gewerbliche und die schnellste Open-Source Nachrichtenorientierte Verteilungsplattform den Komponenten der Dissertation gegenübergestellt. Das Verhältnis bei der Geschwindigkeit ist bei den verschiedenen Laufzeittests ähnlich, sodass in **Fehler! Verweisquelle konnte nicht gefunden werden.**, Abbildung 6-17 und Abbildung 6-18 nur ein Laufzeittest mit 10 Klientenpaaren und 1000 Nachrichten pro Klient abgebildet ist.

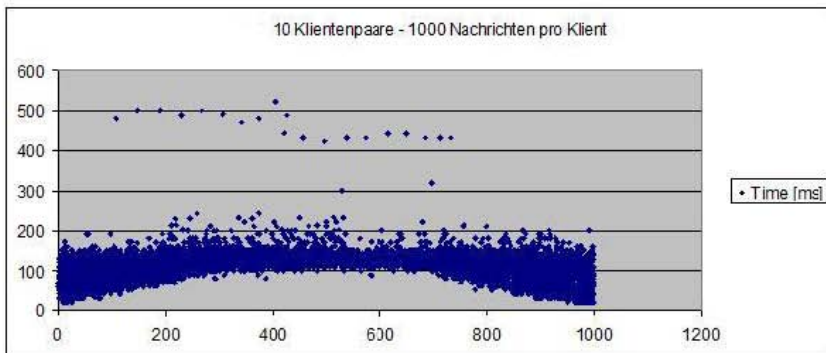


Abbildung 6-16: Vergleichstest - Komponenten der Dissertation

Die durchschnittliche Zustellzeit einer Nachricht von 10 Klienten liegt beim Vergleichstest mit den Komponenten der Dissertation bei 116,5 ms. Bei genau demselben Testdurchlauf hat die schnellste gewerbliche Nachrichtenorientierte Verteilungsplattform in Abbildung 6-17 ersichtlich den zweitschnellsten Wert erreicht.

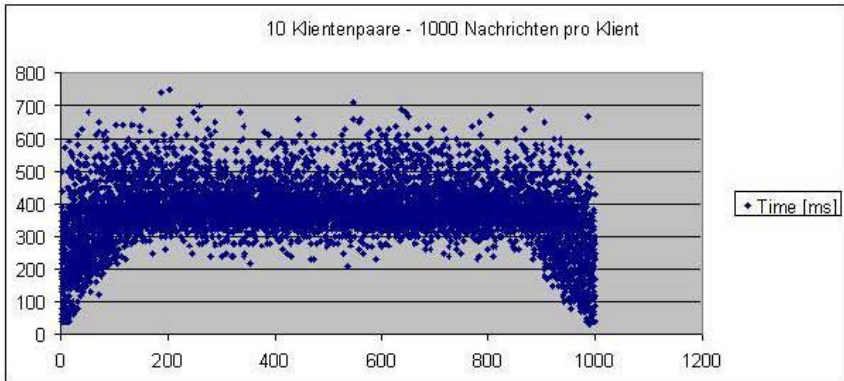


Abbildung 6-17: Vergleichstest - gewerbliche Nachrichtenorientierte Verteilungsplattform

Die durchschnittliche Zustellzeit einer Nachricht von 10 Klienten liegt beim Vergleichstest bei 371,5 ms. Die schnellste Open-Source Nachrichtenorientierte Verteilungsplattform ist in Abbildung 6-18 dargestellt.

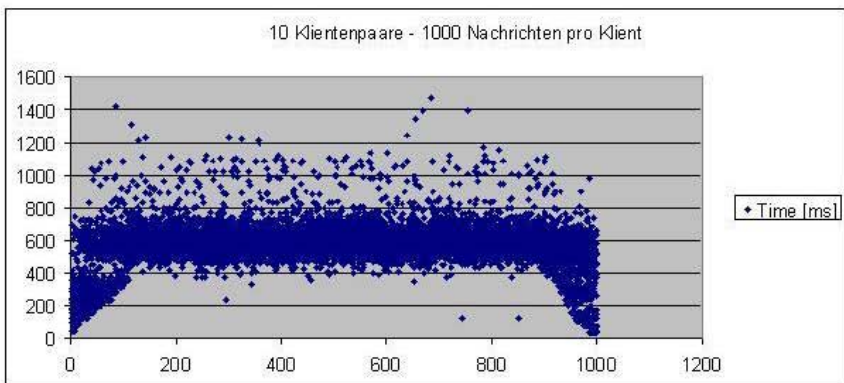


Abbildung 6-18: Vergleichstest - Open-Source Nachrichtenorientierte Verteilungsplattform

Die durchschnittliche Zustellzeit einer Nachricht von 10 Klienten liegt beim Vergleichstest bei 573 ms. Somit können durch die zusätzlichen Komponenten dieser Arbeit Nachrichten schneller als bisher zugestellt werden. Über alle Laufzeittests hinweg ergab sich bei der durchschnittlichen Geschwindigkeit der Komponenten dieser Arbeit eine Zeit von 24ms, gegenüber von 36,9ms bei der gewerblichen und 38,7ms bei der Open-Source Verteilungsplattform. Dies entspricht einem **Geschwindigkeitsgewinn von 65%** gegenüber der schnellsten bekannten Nachrichtenorientierten Verteilungsplattform. Somit wurde die abgeleitete Anforderung

- Punkt-zu-Punkt Nachrichtenübermittlung mit höherem Durchsatz als bei bestehenden Systemen

erfüllt. Der gewonnene Geschwindigkeitsvorteil hat wiederum verschiedene Auswirkungen auf die Produktion. Für den Fall, dass die Nachrichtenübermittlung die Geschwindigkeit des Gesamtsystems vor der Installation der Komponenten dieser Arbeit negativ beeinflusst hatte gibt es folgende Vorteile:

- Es treffen verspätete Nachrichten wieder rechtzeitig ein und ermöglichen somit, dass der Produktionsprozess rechtzeitig gestartet werden kann wodurch eine Anlage schneller produziert und dadurch unter Umständen die gesamte Produktion effizienter wird.
- Die instabile Verbindung zur Anlage wird stabilisiert, wodurch Nachrichten fehlerfrei zugestellt werden können.
- Weitere Anlagen und Applikationen können dem Gesamtsystem hinzugefügt werden, ohne dass die Nachrichtenübermittlungsgeschwindigkeit des Gesamtsystems verringert wird.
- Belastungsreduktion des Netzwerkes.

Vor allem kann die gewonnene Nachrichtenübermittlungs-Zeit für zusätzliche Qualitätssicherung eingesetzt werden. Bei Bedarf können z.B. detailliertere Sensordaten übermittelt werden, die den Produktionsprozess potentiell optimieren oder genauere Daten der Produktionsplanung zustellen können.

6.2 Abbildung von komplexen Beziehungen im Namensraum

In Abbildung 4-5 wurde im UML Diagramm dargestellt, dass der implementierte Namensdienst einen Namensraum als gerichteten, azyklischen Graph aufspannt. In Kapitel 3.5.1 und 4.1.2 wird belegt, dass durch einen gerichteten, azyklischen Graph komplexe Beziehungen direkt in linearer Geschwindigkeit abgebildet werden.

Die Produktion profitiert von der Erfüllung dieser Anforderung dadurch, dass der Aufwand der Abbildung der Produktion als Datenhierarchie und der Aufwand zur Wartung der Daten-Darstellung verringert wird.

6.3 Einbeziehung aller Informationen des Empfängers für eine situationsspezifische Anpassung der Nachricht an das optimale Nachrichtenprotokoll des Empfängers

Die Anpassung der Nachricht an das Empfangsmedium wird durch die Komponente Routenplanung erreicht und wurde in Kapitel 4.3 detailliert beschrieben. Die Routenplanung besitzt zur Laufzeit alle Informationen über die Verbundenen Klienten. Darüber hinaus werden in einer Zuordnungstabelle Details über die Klienten festgehalten, welche Nachrichteninhalte ausgeführt oder dargestellt werden können.

Durch die Erfüllung dieser Anforderung werden Produktionsdaten möglichst optimal zugestellt, sodass diese sofort, am richtigen Ort und in der richtigen Darstellung Verfügbar sind um Produktionsentscheidungen treffen zu können. Zusätzlich wird die Nachrichtenzustellung für den Benutzer wie z.B. einem Wartungsarbeiter vereinfacht, da er sobald sein präferiertes Ein-/Ausgabegerät eingeschaltet ist automatisch nach der Anmeldung an der Nachrichtenorientierten Verteilungsplattform alle Nachrichten über dieses Gerät erhält.

6.4 Ausführung von Arbeitsabläufen von Sender und Empfänger unter Berücksichtigung der Situation des Gesamtsystems

Bei der schnellen Ausführung von Arbeitsabläufen besitzt eine typische Arbeitsablaufverwaltung immer eine Latenzzeit zum Aufruf dieser Komponente. Durch die in dieser Arbeit entwickelte Ausführung von Arbeitsabläufen für die Nachrichtenzustellung im Namensdienst, wird diese Latenzzeit eingespart. Zusätzlich stehen die situationspezifischen Informationen des Gesamtsystems der Routenplanung zur Verfügung und können während der Ausführung von Arbeitsabläufen verwendet werden. Die Arbeitsablaufabstimmung löst Arbeitsabläufe von Sendern und Empfängern auf und lässt somit die Ausführung einer maximalen Anzahl von Arbeitsabläufen zu.

Durch Arbeitsabläufe können Abläufe in der Produktion automatisiert werden. Eine direkte Integration von Kundenwünschen ist somit möglich oder Wartungsangestellte können die Zustellung von Wartungs- oder Alarm-Nachrichten positiv beeinflussen.

7 Zusammenfassung und Ausblick

In der Produktion werden durch eine steigende Anzahl von Anlagen und Applikationen, einem steigendem Automatisierungsgrad und einem wachsenden Detaillierungsgrad der zu versendenden Daten, **schnelle Nachrichtenorientierte Verteilungsplattformen** notwendig, um den bestehenden Nachrichtendurchsatz zu bewältigen. Bestehende Nachrichtenorientierte Verteilungsplattformen in der Produktion besitzen entweder einen Nachrichtendurchsatz von über tausend Nachrichten pro Sekunde oder eine flexible und hochdynamische Nachrichtenzustellung. In dieser Arbeit wird mit einem geschlossenem Ansatz ein einheitliches Systemkonzept für schnelle Basiskomponenten einer Nachrichtenorientierten Verteilungsplattformen entwickelt, die sich zudem **flexibel und hochdynamisch an die Situation des Gesamtsystems anpassen** und Arbeitsabläufe für die Nachrichtenübermittlung ausführen. Durch neue Konzepte in der Nachrichtenübermittlung und **Arbeitsablauf-Ausführung** werden in dieser Arbeit innovative Komponenten einer Nachrichtenorientierten Verteilungsplattformen entwickelt.

Im **Stand der Technik** wurden Verteilungsplattformen und deren Basis-Komponenten betrachtet. Nach einer Bewertung des Stands der Technik wurden Anforderungen an eine schnelle und hochdynamische Nachrichtenorientierte Verteilungsplattform abgeleitet. Keine der am Markt verfügbaren Nachrichtenorientierten Verteilungsplattformen konnte alle Anforderungen erfüllen, weshalb neue innovative Konzepte und Komponenten benötigt werden.

Die Anforderungen an eine schnelle und hochdynamische Nachrichtenorientierte Verteilungsplattform wurden in der **Konzeption** detailliert. Die Konzepte führten zu einer neuen Aufteilung der Komponenten einer Nachrichtenorientierten Verteilungsplattform.

Darauf aufbauend wurden die Komponenten für die Nachrichtenorientierte Verteilungsplattform **entwickelt**. Durch die Komponenten Namensdienst, Routenplanung und Arbeitsablaufabstimmung wurde die Basis für eine Nachrichtenorientierte Verteilungsplattform geschaffen.

Durch den Aufbau des Namensraums als azyklischer Graph, ist es möglich komplexe Beziehungen abzubilden. Der **Namensdienst** sorgt für eine schnellere Nachrichtenübermittlung, indem Namen schneller aufgelöst werden. Zusätzlich können Arbeitsabläufe für die Nachrichtenzustellung innerhalb von Profilen im Namensraum des Namensdienstes definiert werden, die schneller als in herkömmlichen Arbeitsablaufverwaltungen ausgeführt werden. Durch die Eigenentwicklung des Namensdienstes wurde Abstand von standardisierten Namensbäumen genommen, um eine größere Flexibilität in der Gestaltung zu haben und Graphen aufbauen zu können. Dieser Abstand von der Standardisierung hat aber auch einen Nachteil. So können vorhandene Namensräume aus anderen Nachrichtenorientierten Verteilungsplattformen nicht einfach übernommen werden, sondern müssen neu modelliert werden. Genauso kann man die schon mit der Namensdienst-Komponente der Dissertation abgebildeten Namensräume nicht in andere Systeme übernehmen. Die einzige Möglichkeit zur Übernahme wäre, den kompletten Namensdienst in eine andere Nachrichtenorientierte Verteilungsplattform zu transferieren. Somit hat die Flexibilität und Schnelligkeit auch eine Kehrseite und muss mit der Anpassung an Standards abgewogen werden.

Die Komponente **Routenplanung** beinhaltet eine Zuordnungsverhandlung, die eine Nachricht situationsspezifisch an den optimalen Empfänger anpasst. In einer Zuordnungstabelle werden alle Informationen über Klienten im System zur Laufzeit eingesammelt und bilden die Entscheidungsbasis für eine flexible und hochdynamische Nachrichtenübermittlung. Die Anpassung der Nachricht an das Empfangsmedium ist von großem Vorteil, wenn eine Person sofort benachrichtigt werden soll. Abhängig davon, welches Empfangsgerät die Person zuerst erreicht, wird dieser Weg der Nachrichtenübermittlung gewählt. Wenn z.B. ein Wartungsarbeiter gerade mit einem mobilen Gerät unterwegs ist, bekommt er die Nachricht direkt dorthin zugestellt. Ein Problem, das hier auftauchen kann ist, dass das Empfangsgerät nicht in der Lage ist, die Nachricht komplett abzubilden. So kann z.B. eine graphische Auswertung der Produktivität einer Anlage nicht auf einem Mobiltelefon dargestellt werden. Die Routenplanung dieser Arbeit muss deswegen die Details jedes Empfangsgerätes kennen, um eine plausible Weiterleitung der Nachricht vorzunehmen. Wenn diese Details nicht im Vorfeld an die Verteilungsplattform gemeldet werden, ist die Nachrichtenorientierte Verteilungsplattform nicht in der Lage, zu entscheiden, welches

Empfangsgerät das am besten Geeignete ist. Wenn das Empfangsgerät einer Person nicht geeignet ist, die Nachricht darzustellen, so wird zumindest eine kurze Benachrichtigung auf das Empfangsgerät zugestellt, dass eine neue Nachricht verfügbar ist.

Die Komponente **Arbeitsablaufabstimmung** bietet eine regelbasierte Auflösung von widersprüchlichen Arbeitsabläufen. Wenn widersprüchliche Arbeitsabläufe dennoch nicht durch konfigurierte Regeln und Prioritäten aufgelöst werden können, wird eine Verhandlung der Arbeitsabläufe initiiert. Die Verhandlung löst widersprüchliche Arbeitsabläufe auf und sorgt dafür, dass eine möglichst hohe Anzahl an Arbeitsabläufen fehlerfrei ausgeführt wird. Bei der schnellen Behandlung und Ausführung von Arbeitsabläufen ist die Arbeitsablaufabstimmung nur auf Arbeitsabläufe innerhalb der Nachrichtenzustellung ausgelegt. Arbeitsabläufe von Sendern und Empfängern werden aufgelöst und lassen somit die Ausführung einer maximalen Anzahl von Arbeitsabläufen zu. Innerhalb von einem Enterprise Service Bus ist es mittlerweile möglich, allgemeine Formen von Arbeitsabläufen aufzulösen, wobei hier die Latenzzeit für die Ausführung der allgemeinen Arbeitsabläufe hoch ist, da man eine eigene Komponente dafür benötigt. Die Ausführung von Arbeitsabläufen innerhalb der Namensauflösung speziell für die Nachrichtenzustellung spart sich diese Latenzzeit und ist somit schneller.

In der **Realisierung** wurde die Implementierung der Komponenten der Nachrichtenorientierte Verteilungsplattform beschrieben. Diese Komponenten wurden innerhalb von Prototypen in verschiedenen Industrien angewendet. Durch eine Bewertung der Anwendungen wird bewiesen, dass die Anforderungen an eine schnelle und hochdynamische Nachrichtenorientierte Verteilungsplattform erfüllt werden. Eine Geschwindigkeitssteigerung gegenüber vorhandenen Nachrichtenorientierten Verteilungsplattformen von 65% wurde durch Laufzeittests ermittelt.

7.1 Grundlagen zur Weiterentwicklung

Entwicklungsbedarf besteht bei der Zusammenführung einer schnellen Nachrichtenübermittlung und allgemeinen Arbeitsablaufverwaltungen. In dieser Arbeit wurden

Arbeitsabläufe betrachtet, die speziell auf die Nachrichtenübermittlung ausgelegt sind. Hochkomplexe Arbeitsabläufe können innerhalb von Profilen im Namensraum nicht dargestellt und ausgeführt werden. Deshalb wäre eine Integrationsplattform wünschenswert, die verschiedene Arbeitsablaufverwaltungen steuert. Konzepte einer Service Oriented Architecture (SOA) [Barry03, Spath06b] sind darauf ausgelegt, solch eine Integrationsplattform zu entwickeln. Eine Integration von **BPM** mit einer **Business Process Engine** (BPE) zur Ausführung von Arbeitsabläufen kann hier sinnvoll sein.

Die Komponente Namensdienst erreicht eine schnellere Namensauflösung durch selbstentwickelte Klassen und Objekte. Dadurch ist der Namensdienst nicht mehr **standardisiert** und es können keine Daten direkt aus bestehenden Namensräumen übernommen werden. Auch kann der mit dieser Komponente aufgebaute Namensraum nicht direkt in einen anderen Namensdienst einer Nachrichtenorientierten Verteilungsplattform kopiert werden.

Die Arbeitsablaufabstimmung wurde als regelbasierte Komponente entwickelt und implementiert. Die Verhandlungen von Arbeitsabläufen zur Auflösung deuten darauf hin, dass diese Komponente durch einen **Agentenbasierten Ansatz** mehr Komplexität erreichen würde. Damit wäre es möglich, allgemeine Arbeitsabläufe aufzulösen. Innerhalb des Forschungsprojektes Abilities wird genau dieser Ansatz verfolgt, um bis zum Jahr 2008 eine neue Komponente innerhalb einer Föderierten Architektur zu entwickeln.

8 Summary

Message oriented middleware is becoming more and more important in manufacturing and automation. Due to the ever-expanding size of production plants, the rising number of equipment required and the increasing degree of automation, manufacturing is becoming more and more complex [Dorf94, ITRS04, West06]. As automation levels (right up to full automation) increase, all the clients involved have to be able to communicate with one another inside the production plant, e.g. equipment, devices, applications, software systems, data bases, PC's and mobile devices. Message oriented middleware (MOM) ensures that communications are established and standardized between the various clients. Data sent within a production plant is also becoming more and more complex with the result that a higher density of communication takes place with larger and an increased number of messages. New standardization measures taken in equipment communication such as XML [Bray06] will increase the density of communications even more because of the amount of formatting information which needs to be sent [Sutter05, Dreiss05, Meier06]. For example, in the semiconductor industry, approximately 600 pieces of equipment and 1000 clients are required to manufacture 300 mm wafers in a production hall and need to send messages to each another every second [Osborne06, Mönch03, M2A03a]. In the memory and microchip manufacturing industries, this results in a throughput of several thousand messages per second which need to be sent and processed simultaneously with a consistently high level of service and error-free delivery.

Message-oriented middleware not only links up a number of pieces of equipment to one another but also to a multitude of applications such as manufacturing execution systems, enterprise resource planning (ERP), computer aided design (CAD) and production planning and simulation.

High communication densities have different effects on manufacturing. A few examples are mentioned below. Equipment control commands may arrive late and therefore delay production. The connection to a piece of equipment may fail with the risk of machine downtime. It could be that some applications can no longer be utilized because the

overall system would become unstable. A constant overload of the overall system would lead to error-proneness, possibly resulting in a server crashing and shutting down production completely.

Only a few middlewares on the market are able to meet the increasing demands of the manufacturing industry. The disadvantage of these middlewares is that their high speed results in them not being sufficiently flexible and dynamic. This paper is therefore concerned with designing and implementing message-oriented middleware components capable of fulfilling the requirements of a fast and highly dynamic communications system.

Flexible and highly-dynamic message delivery enables a message to dynamically map the state of the overall system. This ensures that an operator best-suited for the task is informed about an equipment shutdown when it occurs. The operator automatically receives the message with the input device which it is currently communicating with the system and is available.

The importance of message-oriented middlewares is substantiated by market observations in the field of software development. According to Gartner at Dataquest, message-oriented middleware is the most-promising market in the field of middlewares and enterprise application integration (EAI) [Tolle07, Meyer02, Ruh01] with a profit increase of 160.7% and a turnover of 8.5 billion dollars in 2005 [Gartner06].

The state of the art of middlewares and their basic components has been studied. Requirements for a fast and highly-dynamic message-oriented middleware were derived on assessing the state of the art. None of the message-oriented middlewares currently available on the market are able to fulfill all the demands made which is why there is a need for new innovative concepts and components.

The detailed requirements of a fast and highly-dynamic message-oriented middleware were considered in a design phase. This showed that point-to-point message transmission is required with a higher throughput than that of existing systems. The naming service has to be capable of spanning a naming space as an anti-cyclical graph. The demand is for rapid execution of workflows in the naming service during naming

resolution, the resolution of competing workflows and situation-related workflow reconciliation. There is also a need to incorporate all recipient information in order to enable the situation-related mapping of messages to an optimum recipient message protocol.

This formed the basis for developing message-oriented middleware components. The components themselves, i.e. naming service, routing and workflow reconciliation, then formed the basis of the message-oriented middleware. Complex relationships can be depicted by designing the naming space in the form of as an acyclic graph. The function of the naming service is to speed up message transmission by resolving names faster. Additionally, workflows for message delivery can be defined within profiles in the naming space of the naming service which are executed faster than in conventional workflow management. The routing component includes an allocation negotiation which maps a message to the optimum recipient according to the situation at the time. All information about clients in the system is collected in an allocation table for the runtime elapsed and forms the basis for decision-making to enable flexible and highly-dynamic message transmission. The workflow reconciliation component provides the control-based resolution of conflicting workflows. If conflicting workflows still cannot be resolved using configured rules and priorities, a negotiation is initiated. The negotiation resolves conflicting workflows and ensures that the maximum number of workflows is executed error-free.

The dependencies, relationships and inheritances of the various message-oriented middleware components are shown in Figure 8-1.

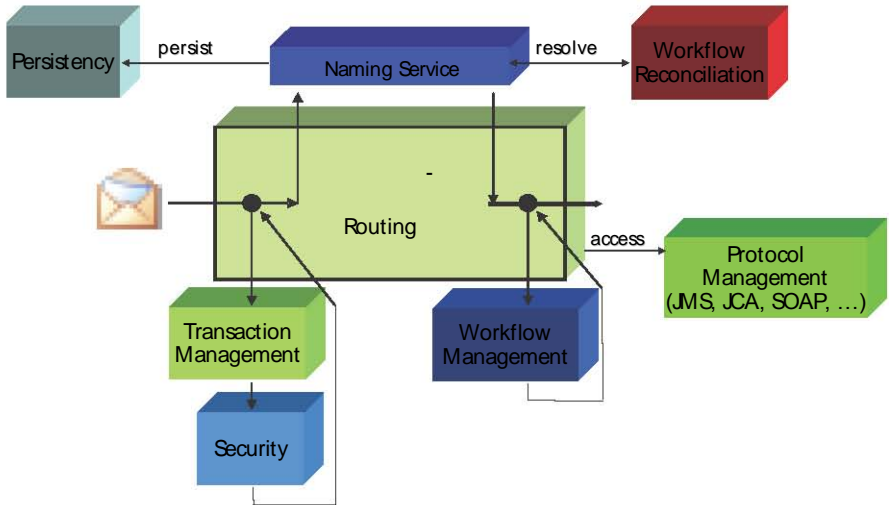


Figure 8-1: Dependency between the components of message oriented middleware

The main component of the message-oriented middleware is routing. It accepts incoming messages and manages them. It determines in which sequence the other components will receive the message and requests the naming service to resolve names. Only if routing and naming service cooperate with one another is it possible for routing to convert the message in the request-response protocol based on the type of address saved in the name graphs. As workflow reconciliation monitors and coordinates the execution of workflows in profiles in the naming service and in the workflow management component, the naming service and workflow reconciliation components need to be integrated together in order to safeguard that conflicting workflows are resolved. The name tree is persisted by the persistency component. Routing has direct access to protocol management to convert the message in the protocol of the recipient. The message is then forwarded at extension points to other components of the message-oriented middleware.

Implementation of the components making up the message-oriented middleware has been described in the section on realization. These components were applied as prototypes in various industries. An evaluation of the applications demonstrated that the requirements of a fast and highly-dynamic message-oriented middleware were fulfilled.

An increase in speed of 65% was recorded during performance tests compared with the speeds of existing message-oriented middlewares.

A method needs to be developed to combine rapid message transmission with general workflow management. In this paper, workflows have been considered which are specially designed for message transmission. Highly complex workflows can not be represented and executed within profiles in the naming space. It would be therefore be desirable to have an integration platform for controlling the various workflow management tasks. Service oriented architecture (SOA) [Barry03, Spath06b] concepts are constructed for developing such an integration platform. Here it would make sense to integrate BPM with a business process engine (BPE) for workflow execution.

9 Literatur

- [Aalst02] Aalst, W.v.; Hee, K.v.:
Workflow Management - Models, Methods, and Systems.
Cambridge: The MIT Press, 2002
- [Aguilera99] Aguilera, M.K. u.a.:
Matching events in a content-based subscription system.
In: ACM: Proceedings of the Eighteenth ACM Symposium on
Principles of Distributed Computing. May 3 - 6, 1999, Atlanta,
Georgia, USA. New York: ACM Press, 1999, S. 53-61, 1999
- [Al03] Al-Najjar, B.; Alsyouf, I.:
Selecting the most efficient maintenance approach using fuzzy
multiple criteria decision making.
In: International Journal of Production Economics 84 (2003), Nr.
1, S. 85-100
- [Allen01] Allen, R.:
Workflow: An Introduction.
Extracted from the Workflow Handbook 2001, S.15-38
Workflow Management Coalition Public Document
[http://www.wfmc.org/standards/docs/Workflow_An_Introduction.
pdf](http://www.wfmc.org/standards/docs/Workflow_An_Introduction.pdf) (20.04.2007)
- [Alonso04] Alonso, G. u.a.:
Web Services.
Berlin u.a.: Springer, 2004
- [Alonso05] Alonso, G.:
Middleware 2005: ACM, IFIP, USENIX 6th International Mid-
dleware Conference. Proceedings. November 28 - December 2,

2005, Grenoble, France.
Berlin: Springer, 2005

- [Ander07] Andersen, E.:
The X.500 Directory - The mother of all directories.
In: Andersen, E.:
The wonderful world of directories
<http://home20.inet.tele.dk/era/x500/x500-technology.htm>
(24.05.2007)
- [Arnold95] Arnold, O. u.a.:
Virtuelle Unternehmen als Unternehmenstyp der Zukunft?
In: Handbuch der maschinellen Datenverarbeitung 185 (1995), S.
8-23, 1995
- [Balzert01] Balzert, H.:
Lehrbuch der Software-Technik. Bd.1. Software-Entwicklung,
2. Aufl.
Heidelberg: Spektrum Akademischer Verlag, 2001
- [Barry03] Barry, D.K.:
Web Services and Service-Oriented Architectures: The Savvy
Manager's Guide.
San Francisco u.a.: Morgan Kaufmann, 2003
- [Berger05] Berger, R.:
Sicherheitszentrierte Architektur für Internet-basierte Dienste im
Maschinen- und Anlagenbau.
Stuttgart : Fraunhofer-IRB-Verl., 2005
(Berichte aus dem Produktionstechnischen Zentrum)
Zugl. Berlin, Univ., Diss., 2005

- [Bray06] Bray, T. u.a.:
Extensible Markup Language (XML) 1.0
W3C Recommendation, 16.08.2006. 4th Edition.
<http://www.w3.org/XML/Core/#Publications/TR/2006/REC-xml-20060816> (23.04.2007)
- [Brom05] Bromberg, Y-D.; Issarny, V.:
INDISS: Interoperable Discovery System for Networked Services.
In: Alonso, G.: Middleware 2005: ACM, IFIP, USENIX 6th International Middleware Conference. Proceedings. November 28 - December 2, 2005, Grenoble, France.
Berlin: Springer, 2005, S. 164-183
- [Cam88] Campbell, J.:
V24/RS-232 Kommunikation.
5. Aufl.
Berkeley u.a.: Sybex, 1988
- [Castro02] Castro, M. u.a.:
SCRIBE: A large-scale and decentralized application-level multicast infrastructure.
In: IEEE Journal on Selected Areas in Communication, 20 (2002) Nr. 8, S. 100-110
- [Cecchet03] Cecchet, E. u.a.:
Performance Comparison of Middleware Architectures for Generating Dynamic Web Content.
In: Endler, M. (Hrsg): Middleware 2003: ACM/IFIP/USENIX International Middleware Conference. Proceedings. Rio de Janeiro, Brazil, June 16 - 20, 2003.
Berlin: Springer, 2003, S. 242-261

- [Cham02] Chamsieva, L.; Geiger, J.; Wieland, M.:
Vergleich von Architekturen zur Gerätevernetzung.
Fachstudie Stuttgart, 2002.
<http://nbn-resolving.de/urn:nbn:de:bsz:93-opus-25704>
(23.04.2007)
- [Chap04] Chappell, D.A.:
Enterprise Service Bus.
Beijing: O'Reilley, 2004
- [Cheung06] Cheung, A.K.Y.; Jacobsen, H.-A.:
Dynamic Load Balancing in Distributed Content-Based Publish/Subscribe.
In: Stehen, M.v. (Hrsg); Henning, M. (Hrsg):
Middleware 2006: ACM, IFIP, USENIX 7th International Middleware Conference. Proceedings. Melbourne, Australia, November 27 - December 1, 2006.
Berlin: Springer, 2006, S. 141-161
- [Choksi01] Chokai, A.:
Hierarchical Routing in Sensor Networks, CS-672: Seminar on Pervasive and Peer-to-Peer Computing, Storage & Networking. Term-Paper Submission, Rutgers University, 2001.
- [CORBA04] An Adopted Specification of the Object Management Group, Inc:
Common Object Request Broker Architecture: Core Specification (March 2004)
http://www.omg.org/technology/documents/corba_spec_catalog.htm
(07.04.2007)
- [Diestel00] Diestel, R.:
Graphentheorie.

2., neu bearb. und erw. Aufl.
Berlin; Heidelberg u.a.: Springer, 2000.

- [Degen01] Degenaro, L.; Iyengar, A.; Rouvellou, I.:
Improving Performance with Application-Level Caching.
In: Proceedings of the SSGR 2001 International Conference
on Advances in Infrastructure for Electronic Business,
Science and Education on the Internet, L'Aquila, Italy, August 6 -
12, 2001
<http://www.research.ibm.com/people/i/iyengar/ssgrr01.pdf>
(19.04.2007)
- [Desfo06] Desforges, X.; Archimede, B.:
Multi-agent framework based on smart sensors/actuators for ma-
chine tools control and monitoring
In: Engineering Applications of Artificial Intelligence 19 (2006)
Nr. 6, S. 641-655
- [Dorf94] Dorf, R.C.; Kusiak, A. (Eds.):
Handbook of Design, Manufacturing, and Automation.
New York u.a.: Wiley, 1994.
- [Dreiss05] Dreiss, P.:
Flexible Manufacturing Application Integration (MAI) in Photo-
voltaic Production Environments : A service based Manufacturing
Environment.
In: Dorner, Johann (Leitung); Fraunhofer-Institut für Produktion-
stechnik und Automatisierung IPA: Advanced Production Tech-
nologies in the Photovoltaic Industries: Manufacturing concepts
and their transferability. 27. September 2005, Stuttgart.
Stuttgart, 2005, 27 S.

- [Emm00] Emmerich, W.:
The Future of Software Engineering.
In: Finkelstein, A. (Ed): The Future of Software Engineering: 22nd
International Conference on Software Engineering.
New York: ACM Press, 2000, S. 117 – 129
- [Endler03] Endler, M.:
Middleware 2003: ACM/IFIP/USENIX International Middleware
Conference. Proceedings. Rio de Janeiro, Brazil, June 16 - 20,
2003.
Berlin: Springer, 2003
- [Feldmann98] Feldmann, K.; Rottbauer, H.; Stöckel, T.:
Information Systems Architecture for Collaborative Manufacturing
in Virtual Enterprises.
In: Jacucci, G. (Ed.): Globalization of manufacturing in the digital
communications era of the 21st Century. Proceedings of the 10th
International IFIP WG5.2/5.3 International Conference
PROLAMAT 98, Trento, Italy, September 9 - 11 & 12, 1998.
Boston: Kluwer, 1998, S. 711-724
- [Ford05] Ford G.H.:
Security issues for middleware and web services : design practices,
standards and metrics.
Towson: 2005
- [Foster04] Foster, I.; Kesselman, C.:
The Grid Blueprint for a New Computing Infrastructure.
2. Auflage
Amsterdam u.a.: Morgan Kaufmann, 2004
- [Gartner02] Correia, J.M.:
Application Integration, Middleware and Portals Market Statistics.

Stamford, USA: Gartner, 2002 (Gartner Market Statistics)

[Gartner05] Gartner says EMEA Application Integration and Middleware Market is Set to Grow Seven Percent in 2005 but the Industry Outlook Remains Challenging.
Gartner Press Release. Stamford, 2005
<http://www.gartner.com/it/page.jsp?id=492155> (30.04.2007)

[Gartner06] Correira, J.M.; Biscotti, F.:
Market Share: AIM and Portal Software, Worldwide, 2005.
Stamford: Gartner, 2006

[Gartner07] Cantara, M. u.a.:
Forecast: Portal, Process and Middleware Software, Worldwide, 2006-2011, Update
Stamford, USA: Gartner, 2007 (Gartner Market Statistics)

[Gray93] Gray, J.; Reuter, A.:
Transaction Processing: Concepts and Techniques
San Mateo: Morgan Kaufmann, 1993

[Gribble01] Gribble, S.D. u.a.:
The Ninja Architecture for Robust Internet-Scale Systems and Services.
In: Computer Networks –The International Journal of Computer and Telecommunications Networking 35 (2001), Nr. 4, S. 473-497

[GSSAPI02] Sun Microsystems:
GSS-API Programming Guide.
Santa Clara, 2002
<http://192.18.109.11/816-1331/816-1331.pdf> (10.05.2007)

- [Gupta04] Gupta, A. u.a:
Meghdoot: Content-based Publish/Subscribe over P2P Networks.
In: Jacobsen, H.-A.:
Middleware 2004: ACM, IFIP, USENIX International Middleware
Conference. Proceedings. Toronto, Canada, October 18 - 20, 2004.
Berlin: Springer, 2004, S. 254-273
- [Gurz03] Gurzki, T.; Hinderer, H.:
Eine Referenzarchitektur für Software zur Realisierung von Unter-
nehmensportalen.
In: Reimer, U.; (Hrsg.) u.a.:
WM2003: Professionelles Wissensmanagement - Erfahrungen und
Visionen : Beiträge der 2. Konferenz. 2. - 4. April in Luzern,
Schweiz (Lecture Notes in Informatics (LNI) - Proceedings; 28).
Bonn : Ges. für Informatik, 2003, S. 157-160
- [Ha05] Hammerschall, U.:
Verteilte Systeme und Anwendungen - Architekturkonzepte, Stan-
dards und Middleware-Technologien.
München: Pearson, 2005
- [Han06] Han, T.; Yang, B-S.:
Development of an e-maintenance system integrating advanced
techniques
In: Computers in Industry 57 (2006), Nr. 6, S.569-580
- [Hall80] Hall, S.:
Encoding/decoding.
In: Centre for Contemporary Cultural Studies (Hrsg.):
Culture, Media, Language: Working Papers in Cultural Studies,
1972-79.
London: Hutchinson, 1980, S. 128-38

- [Hall97] Hall, S.:
The Television Discourse-Encoding and Decoding.
In: Gray, A.; McGuigan, J. (Hrsg):
Studying Culture: An Introductory Reader, Second Edition
London: Arnold, 1997, S. 28-34
- [Harrop05] Harrop, R.; Machacek, J.:
Pro Spring.
Berkeley: APress, 2005
- [Heer99] Heering, H.:
Objektorientierte Middleware-Konzepte und Infrastrukturen in ver-
teilten Anwendungen.
Hamburg: Diplomata, 1999
- [Heide01] Heidemann, J. u.a.:
Building Efficient Wireless Sensor Networks with Low-Level
Naming.
In: Proceedings of the ACM Symposium on Operating System
Principles, Chateau Lake Louise, Banff, Alberta, Canada, ACM.
October 21-24 2001, S. 146-159.
<http://www.isi.edu/~johnh/PAPERS/Heidemann01c.html>
(17.07.2007)
- [Heinzel00] Heinzelman, W.; Chandrakasan, A.; Balakrishnan, H.:
Energy-Efficient Communication Protocol for Wireless Microsen-
sor Networks.
In: Sprague, R. H. (Ed.): Proceedings of the 33rd Hawaii
International Conference on System Sciences (HICSS '00), Janu-
ary 4 - 7, 2000, Maui, Hawaii, S. 8020
- [Hiber07] Red Hat:
Object and Relational Mapping (ORM) with Hibernate.

<http://www.jboss.com/pdf/HibernateBrochure-Jun2005.pdf>
(22.05.2007)

- [Hinden06] Hinden R.; Deering, S.:
RFC 3513: Internet Protocol Version 6 (IPv6) Addressing Architecture.
Internet RFCs, 2003
<http://portal.acm.org/citation.cfm?id=RFC3513> (22.05.2007)
- [Howe07] Howe, D.:
The Free On-line Dictionary of Computing.
<http://foldoc.org/> (02.04.2007)
- [Wack97] Wackerow, Dieter; IBM:
Internet Application Development with MQSeries and Java.
Vervante Corporate Publishing, 1997
- [IFX05] Linde, H.; Böstler, F.:
Infineon, Dresden.
Gespräch: Ausfallszenarien in der Halbleiterproduktion.
03.03.2005, 12:00 Uhr, Dresden
- [Ihns03] Ihns, O. (Hrsg.) u.a.:
Enterprise JavaBeans komplett.
München; Wien: Oldenbourg, 2003
- [INS99] OMG TC Document ptc:
Interoperable Naming Service.
09.09.1999
- [Intana00] Intanagonwiwat, C.; Govindan, R.; Estrin, D.:
Directed diffusion: a scalable and robust communication paradigm
for sensor networks.

In: Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking, August 6-11, 2000, Boston, Massachusetts.
New York: ACM, 2000, S. 56-67

- [ITRS04] International Technology Roadmap for Semiconductors (ITRS); European Semiconductor Industry Association u.a.:
Factory Integration, 2004 Update.
<http://www.itrs.net/Links/2004Update/2004Update.htm>
(08.12.2006)
- [JAAS01] Sun Microsystems:
Java™ Authentication and Authorization Service Reference Guide for the J2SE Development Kit 5.0. (08.08.2001)
<http://java.sun.com/products/jaas/> (10.05.2007)
- [Jacobsen04] Jacobsen, H.-A.:
Middleware 2004: ACM, IFIP, USENIX International Middleware Conference. Proceedings. Toronto, Canada, October 18 - 20, 2004.
Berlin: Springer, 2004
- [JBoss07] Red Hat:
JBoss Application Server.
<http://labs.jboss.com/jbossas/> (03.05.2007)
- [JBossMQ07] Red Hat:
JBoss Messaging.
<http://labs.jboss.com/jbossmessaging/> (03.05.2007)
- [JCE04] Sun Microsystems:
Java™ Cryptography Extension (JCE) Reference Guide for the Java™ 2 Platform Standard Edition Development Kit (JDK) 5.0.

29.01.2004

<http://java.sun.com/products/jce/> (10.05.2007)

[JGroups07]

Red Hat:

Java Groups.

<http://labs.jboss.com/jgroups/> (03.05.2007)

[Jin03]

Jin, J.; Nahrstedt, K.:

Large-Scale Service Overlay Networking with Distance-Based Clustering.

In: Endler, M. (Hrsg): Middleware 2003: ACM/IFIP/USENIX International Middleware Conference. Proceedings. Rio de Janeiro, Brazil, June 16 - 20, 2003.

Berlin: Springer, 2003, S. 394-413

[Jin04]

Jin, J.; Nahrstedt, K.:

On Exploring Performance Optimizations in Web Service Composition.

In: Jacobsen, H.-A. (Hrsg):

Middleware 2004: ACM, IFIP, USENIX International Middleware Conference. Proceedings. Toronto, Canada, October 18 - 20, 2004.

Berlin: Springer, 2004, S. 115-134

[JNDI02]

Sun Microsystems:

Java Naming and Directory Interface™. (2002)

<http://java.sun.com/j2se/1.5.0/docs/guide/jndi/index.html>
(07.04.2007)

[Kaplan01]

Kaplan, G.:

Ethernet's winning ways.

In: IEEE Spectrum 38 (2001), Nr. 1, S. 113-115

- [Klyne04] Klyne, G. (Ed.) u.a.:
Composite Capability/Preference Profiles (CC/PP): Structure and
Vocabularies 1.0: W3C Recommendation 15 January 2004
<http://www.w3.org/TR/CCPP-struct-vocab/> (10.05.2007)
- [Knoll03] Bölstler, F.; Knoll, G.:
Message to anywhere (M2A).
In: Interaktiv 5 (2003), Nr. 2, S. 6-7
- [Knoll04a] Knoll, G.; Dorner, J.; Bölstler, F.; Seidelmann, J.:
Universal and mobile messaging framework M2A "Message to
anywhere" for laboratory automation.
In: Association for Laboratory Automation: LabAutomation 2004:
Final Program & Abstracts. The premier international conference
on laboratory automation. February 1-5, 2004, San Jose, Califor-
nia, USA. San Diego, CA, USA, 2004, S. 140
- [Knoll04b] Knoll, G.; Dorner, J.; Seidelmann, J.:
Universal and Mobile messaging framework M2A "Message to
Anywhere" for Semiconductor Manufacturing.
In: Graf, Otto (Chair); Institute of Electrical and Electronics Engi-
neers u.a.: ASMC 2004: The 15th Annual IEEE/SEMI Advanced
Semiconductor Manufacturing Conference and Workshop, May 4-
6, 2004, Boston, USA. Piscataway, NJ: IEEE, 2004, S. 241-243
- [Knoll04c] Knoll, G.; Jazdi, N.:
M2A - Entwicklung eines universellen und mobilen Kommunika-
tionsframeworks „Message to Anywhere“ in Hightech-Branchen.
InnoNet Projektprofile 2004:
http://www.vdivde-it.de/innonet/projekte/in_pp067_m2a.pdf
(28.04.2007)

- [Knoll05] Knoll, G.:
Universal Communications through Message to Anywhere.
In: International Conference and Exhibition on Drug Discovery
MipTec 2005, May 9-12, 2005, Basel, CH, S. 75
- [Knoll06a] Knoll, G.; Stiedl, T.; Gutbrodt, F.:
Entwicklung eines universellen und mobilen Kommunikationsfra-
meworks "Message to Anywhere" in Hightech-Branchen (M2A):
Abschlussbericht Programm "Förderung von Innovativen Netz-
werken (InnoNet), Förderkennzeichen IN-3522.
Stuttgart: 2006
- [Knoll06b] Bagnato, A.; Guglielmina, C.; Knoll, G.; Tolle, K.:
Federated Message-based Architecture for eBusiness Interoperabil-
ity in New Member States SMEs.
In: Cunningham, Paul (Ed.) u.a.: Exploiting the Knowledge Econ-
omy - Part 1: Issues, Applications and Case Studies. Amsterdam
u.a.: IOS Press, 2006, S. 165-172
(Information and Communication Technologies and the Knowl-
edge Economy, 3)
- [Knoll07] Knoll, G.; Guglielmina, C.; Tolle, K., Kiauleikis, V.:
Interoperability challenges in New Member States Small and Me-
dium Enterprises to derive new architectural concepts.
In: Cardoso, Jorge (Ed.); u.a.; Institute for Systems and Technolo-
gies of Information, Control and Communication u.a.: ICEIS 2007
- Proceedings of the Ninth International Conference on Enterprise
Information Systems - Volume Software Agents and Internet
Computing: Funchal, Madeira, Portugal, June 12 - 16, 2007.
Sebútal, Portugal, 2007, S. 257-260
- [Krogh95] Krogh, M.; Painter, J.; McCormick, J. u.a.:
The ACL Message Passing Library.

Paper submitted to European T3D Workshop, September 7, 1995
<http://www.osti.gov/servlets/purl/109624-0wV3i4/webviewable/>
(23.04.2007).

- [Kru99] Kruchten, P.:
The Rational Unified Process.
Reading, Mass.: Addison-Wesley, 1999
- [Lawrenz00] Lawrenz, W.:
CAN Controller Area Network – Grundlagen und Praxis.
Heidelberg: Hüthig, 4., überarb. Aufl., 2000
- [Lee06] Lee, K.C.; Lee, S.; Lee, M.H.:
Worst case communication delay of real-time industrial switched
ethernet with multiple levels
In: IEEE Transactions on Industrial Electronics 53 (2006) Nr. 5, S.
1669-1676
- [Leser07] Leser, U.; Naumann, F.:
Informationsintegration – Architekturen und Methoden zur Integ-
ration verteilter und heterogener Datenquellen.
Heidelberg: dpunkt.verlag, 2007
- [M2A03a] Böstler, F.; Serrano, J.; Stiedl, T.; Lang, T.; Raab, H.; Helmstreit,
H.; Kugler, E.; Petzold, F.; Ulbricht, F.:
Systema GmbH, Dresden.
Gespräch: Kommunikationsaufkommen in der Halbleiterindustrie.
13.08.2003, 11:00h, Dresden
- [M2A03b] Böstler, F.; Serrano, J.; Stiedl, T.; Lang, T.; Raab, H.; Helmstreit,
H.; Kugler, E.; Petzold, F.; Ulbricht, F.:
Systema GmbH, Dresden.
Gespräch: Einsatz von Publish/Subscribe in der Halbleiterindust-

rie.

14.08.2003, 09:30h, Dresden

- [M2A03c] Jazdi, N.; Serrano, J.; Stiedl, T.; Lang, T.; Helmstreit, H.; Kugler, E.; Frauenhoffer, F.; Steblotsev, K.:
IAS, Universität Stuttgart.
Gespräch: Anforderungen an eine innovative Kommunikation.
07.10.2003, 14:00h, Stuttgart
- [M2A04] Jazdi, N.; Serrano, J.; Stiedl, T.; Gutbrodt, F.; Lang, T.; Petzold, F.; Meier, M.; Wiechers, A.; Frauenhoffer, F.:
IAS, Universität Stuttgart.
Gespräch: Routing innerhalb von Message Oriented Middlewares.
10.02.2004, 10:30h, Stuttgart
- [Mahm04] Mahmoud, Q.H.:
Middleware for communications.
Chichester: Wiley, 2004
- [Mauerer05] Mauerer J.:
ADO.NET 2.0, 19.05.2005.
<http://www.microsoft.com/germany/msdn/library/net/ADONET20.msp?mfr=true> (07.04.2007)
- [Meier06] Meier, M; Dreiss, P.; Muckenhirn, R.:
Building a web service technology-based factory IT-landscape using Web-Objects.
In: Worldwide Developer Conference (WWDC) 2006, 07. - 11. August, San Francisco: 2006, 1S.
- [Mertens94] Mertens, P.:
Virtuelle Unternehmen.
In: Wirtschaftsinformatik 36 (1994) Nr. 2, S. 169-172

- [Meyer02] Meyer, M.; Weingärtner, S.:
Enterprise Application Integration – Grundlagen
In: Meyer, M. (Hrsg.): CRM-Systeme mit EAI: Konzeption, Implementierung und Evaluation. Braunschweig; Wiesbaden: Vieweg, 2002, S. 199-229
- [Mobley90] Mobley, R.K.:
An Introduction to Predictive Maintenance.
New York: Van Nostrand Reinhold, 1990
- [Mock87] Mockapetris, P.V.:
RFC 1035: Domain Names – Implementation and Specification.
Internet RFCs, 1987
<http://portal.acm.org/citation.cfm?id=RFC1035> (11.05.2007)
- [Mönch03] Mönch, L.; Stehli, M.; Zimmermann, J.:
FABMAS: An agent-based system for production control of semiconductor manufacturing processes
In: Holonic and Multi-Agent Systems for Manufacturing, Lecture Notes in Artificial Intelligence, 2744 (2003), S. 258-267
- [Muir07] Muir, J.:
Microsoft Unified Communications Technologies For the Manufacturing Industry.
Microsoft Office Publication, March 2007
<http://www.microsoft.com/office/showcase/manufacturingintel> (12.07.2007)
- [Mul94] Mullender, S.J.:
Distributed Systems.
2. Ed., repr.
New York: acm press, 1994

- [Osborne06] Osborne, M.:
300mm activity report: December 2005 to March 2006.
In: Semiconductor Fabtech 29 (2006), S. 9-20
- [OSI94] International Standard ISO/IEC7498-1: 1994
Information Technology – Open Systems Interconnection – Basic
Reference Model: The Basic Model
- [Park98] Park, J.H.; Yoon, Y.C.:
An extended TCP/IP protocol for real-time local area network.
In: Control Engineering Practice 6 (1998), Nr. 1, S. 111-118
- [Pend03] Pendleton, M.:
Factory integration "sticky points" - The heir apparent to SECS.
In: Solid State Technology 46 (2003), Nr. 9, S. 92
- [Picot96] Picot, A.; Reichwald, R.; Wigand, R.T.:
Die grenzenlose Unternehmung. 2., aktual. Aufl.
Wiesbaden: Gabler, 1996
- [Ponne02] Ponnekanti, S.R.; Rox, A.:
SWORD: A Developer Toolkit for Web Service Composition.
In: Proceedings. The Eleventh World Wide Web Conference
(Web Engineering Track), Honolulu, Hawaii, 7-11 May 2002
- [Popp00] Popp, M.:
Profibus-DP/DPV1 : Grundlagen, Tipps und Tricks für Anwender.
2., überarb. Aufl.
Heidelberg: Hüthig, 2000
- [Pereira01] Pereira, J. u.a.:
WebFilter: A High Throughput XML-based Publish and Subscribe

System.

In: Proceedings of 27th International Conference on Very Large Data Bases VLDB, Roma, 11-14 September 2001.

Orlando, FL, USA: Morgan Kaufman, S. 723-724, 2001

[Prasad04]

Prasad, S.K. u.a.:

SyD: A Middleware Testbed for Collaborative Applications over Small Heterogeneous Devices and Data Stores.

In: Jacobsen, H.-A. (Hrsg):

Middleware 2004: ACM, IFIP, USENIX International Middleware Conference. Proceedings. Toronto, Canada, October 18 - 20, 2004.

Berlin: Springer, 2004, S. 352-371

[RMI06]

Sun Microsystems:

Java Remote Method Invocation (Java RMI)

Santa Clara, 2006

<http://java.sun.com/javase/6/docs/technotes/guides/rmi/index.html>
(07.04.2007)

[Ruh01]

Ruh W.A.; Maginnis, F.X.; Brown, W.J.:

Enterprise Application Integration.

New York: Wiley, 2001

[Russell06]

Russell, C.:

JSR-000243 Java™ Data Objects 2.0 Specification.

Final Release.

Sun Microsystems, Inc. 2006

<http://jcp.org/aboutJava/Communityprocess/final/jsr243/index.html>
(10.05.2007)

[Schantz03]

Schantz, R.E. u.a.:

Flexible and Adaptive QoS Control for Distributed Real-Time and Embedded Middleware.

In: Endler, M. (Hrsg): Middleware 2003: ACM/IFIP/USENIX International Middleware Conference. Proceedings. Rio de Janeiro, Brazil, June 16 - 20, 2003.
Berlin: Springer, 2003, S. 374-393

[Schmid93]

Schmid, B.:

Nutzen der Ressource Information für das Unternehmen.

In: Erfolgsfaktor Information - Schlüssel zu Wettbewerbsvorsprüngen. 5. Fachinformationskongreß für Unternehmer und Führungskräfte. München, 1993.

München: Bayrisches Staatsministerium für Wirtschaft und Verkehr, 1993

[Schmidt98]

Schmidt, D.C.:

Evaluating Architectures for Multithreaded Object Request Brokers.

In: Communications of the ACM 41 (1998) Nr. 10, S. 54

[Schrö03]

Schröder, A.:

Zusammenfassung der Vorlesung RvS.

03. - 06.03.2003

http://www.a-netz.de/downloads/zusammenfassung_rvs.pdf
(23.04.2007)

[Schulz00]

Schulz, M.; Stanley, T.D.; Renelt, B.; Sturm, R.; Schwertschlager, O.:

Simulation Based Decision Support for Future 300mm Automated Material Handling.

In: Joines, J.A. u.a.: 2000 Winter Simulation Conference: The New Simulation Millenium. Orlando, FL, USA, 10-13 December 2000.
New York: Association for Computing Machinery, 2000, S. 1518-1522

- [SemiE37-06] Norm SEMI E37-0703, int. Standard, 2006:
Specification for Processing Management
- [Ser07] The Apache Software Foundation:
Servicemix (2006)
<http://incubator.apache.org/servicemix/documentation.html>
(28.04.2007)
- [Sezcan07] Sezcan, Y.:
Geschäftsoptimierung durch Einsatz einer Kommunikati-
onsinfrastruktur am Beispiel der Halbleiterindustrie.
Ulm, Univ., Fak. Wirtschaftswiss., Dipl.-Arb. 2007
- [Sieg92] Siegmund, G.:
Grundlagen der Vermittlungstechnik.
2., überarb. und erw. Aufl.
Heidelberg: R.v. Decker, 1992
- [Siegel98] Siegel, J.:
OMG Overview: CORBA and the OMA in Enterprise Computing.
In: Communications of the ACM 41 (1998), Nr. 10, S. 37
- [Spath04] Spath, D. (Hrsg.); Fraunhofer IAO:
Effiziente Portalprojekte in der Unternehmenspraxis.
Zweites Anwenderforum Unternehmensportale, 15. Juni 2004, In-
stitutszentrum Stuttgart der Fraunhofer-Gesellschaft.
Stuttgart: Fraunhofer IRB Verlag, 2004
- [Spath06a] Spath, D. (Hrsg.); Fraunhofer IAO:
Dokumenten- und Workflow-Management in der Praxis 2006:
Prozesse, Informationen und Mitarbeiter erfolgreich integrieren.
Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO,

Stuttgart.
Stuttgart: 2006

- [Spath06b] Spath, D. (Hrsg.); Fraunhofer IAO:
Serviceorientierte Architekturen (SOA):
Tagungsband des Stuttgarter Softwaretechnik-Forums 2006, 8.
November 2006.
Stuttgart: Fraunhofer IRB Verlag, 2006
- [Srin95] Srinivasan R.:
RFC 1831: RPC: Remote Procedure Call Protocol Specification
Version 2.
Internet RFCs, 1995
<http://www.freesoft.org/CIE/RFC/Orig/rfc1831.txt> (17.07.2007)
- [Stall00] Stalling, W.:
Local and Metropolitan Area Networks. 6th ed.
Upper Saddle River, NJ: Prentice Hall, 2000
- [Steen06] Stehen, M.v. (Hrsg); Henning, M. (Hrsg):
Middleware 2006: ACM, IFIP, USENIX 7th International Mid-
dleware Conference. Proceedings. Melbourne, Australia, Novem-
ber 27 - December 1, 2006.
Berlin: Springer, 2006
- [Stiedl04] Stiedl, T.:
M2A – Message to Anywhere: Auswertung der Performancetests.
Stuttgart: 2004
- [Stöckel01] Stöckel, T.:
Kommunikationstechnische Integration der Prozessebene in Pro-
duktionssysteme durch Middleware-Frameworks.
Bamberg: Meisenbach, 2001

Erlangen, Nürnberg, Univ., Diss., 2000

- [Sutter05] De Sutter, R.; Lerouge, S.; De Schrijver, D.; Van de Walle, R.:
Enhancing RSS feeds: eliminating overhead through binary
encoding
In: Third Int. Conf. on Information Technology and Applications,
2005. ICITA 2005.
04-07 July 2005, Sydney. Vol.1.
Los Alamitos: IEEE Computer Soc., 2005, S. 520-525
- [Tan95] Tanenbaum, A. S.:
Distributed Operating Systems.
Upper Saddle River: Prentice-Hall, 1995
- [Tan02] Tanenbaum, A.S.; Steen, M. v.:
Distributed Systems: Principles and Paradigms.
Upper Saddle River: Prentice-Hall, 2002
- [Tanner04] Tanner, A.; Mühl, G.:
A formalisation of message complete publish/subscribe systems.
Berlin, Techn. Univ., 2004
(Forschungsberichte der Fakultät IV - Elektrotechnik und Informa-
tik / Technische Universität Berlin; Bericht-Nr. 2004,11)
- [Tibco03] CS514: Intermediate Course in Computer Systems:
Lecture 4: Sept. 10, 2003
Introduction to Message Oriented Middleware (MOM)
<http://cs.cornell.edu/courses/cs514/2003fa/CS514-fa-lec-04v2.pdf>
(25.05.2007)
- [Tolle07] Tolle, K.; Kiauleikis, V.; Justinaviciene, K.; Knoll, G.; Gug-
lielmina, C.; Arezza, A.:
An Interoperable E-business platform towards better integration of

New Member States SME's.

In: Goncalves, R.J.; Müller, J.P.; Mertins, K.; Zelm, M. (Eds.):
Enterprise Interoperability II – New Challenges and Approaches.
London: Springer, 2007, S. 553 – 556

[Ulmer05]

Ulmer P.:

Anwendungsorientierte Middleware.

Hauptseminar Ansätze für Betriebssysteme der Zukunft.

München, Technische Universität, 2005

[Vlach05]

Vlachakis, J.; Gurzki, T.; Kirchhof, A.:

Marktübersicht Portalsoftware 2005

Stuttgart: Fraunhofer IRB Verlag, 2005

[Walker90]

Walker, J.Q.:

A node-positioning algorithm for general trees.

In: Software – Practice and Experience 20 (1990) Nr. 7, S. 685-705

[Walls05]

Walls, C.; Breidenbach, R.:

Spring in Action.

Greenwich: Manning, 2005

[Wang04]

Wang, J.; Jin, B.; Li, J.:

An Ontology-Based Publish/Subscribe System.

In: Jacobsen, H.-A. (Hrsg):

Middleware 2004: ACM, IFIP, USENIX International Middleware
Conference. Proceedings. Toronto, Canada, October 18 - 20, 2004.

Berlin: Springer, 2004, S. 232 -253

[Weber98]

Weber, M.:

Verteilte Systeme.

Heidelberg; Berlin: Spektrum Akademischer Verlag, 1998

- [Weik02] Weikum, G.; Vossen W.:
Transactional Information Systems.
San Francisco: Morgan Kaufmann, 2002
- [West01] Westkämper, E.; Braatz, A.:
Eine Methode zur objektorientierten Softwarespezifikation von de-
zentralen Automatisierungssystemen mit der Unified Modeling
Language (UML).
In: Automatisierungstechnik at 49 (2001) Nr. 5, S. 225-232
- [West06] Westkämper, E.; Warnecke, H.-J.; Decker, M. (Mitarb.):
Einführung in die Fertigungstechnik. 7., bearb. und erg. Aufl.
Wiesbaden: Teubner, 2006
(Teubner Lehrbuch Maschinenbau)
- [Werth03] Werthschulte, K.:
Integration von heterogenen Bussystemen in die Heimautomatisie-
rung unter Verwendung von Middleware.
München, Techn. Univ., Fak. für Elektrotechnik und Informati-
onstechnik, Diss., 2003
- [Weule92] Weule, H.:
Information als Produktionsfaktor.
In: Görke, W.; Rinisland, H.; Syrbe, M. (Hrsg): Information als
Produktionsfaktor. GI-Jahrestagung, Karlsruhe, 28. September bis
2. Oktober 1992.
Berlin u.a.: Springer Verlag, S. 3-19
(Informatik aktuell)
- [Wiese03] Wiese, H.:
Grundlagen der Kommunikationstechnik.
Skriptum zur Vorlesung.

Esslingen: Fachhochschule Esslingen, Hochschule für Technik
2003

- [William00] Williamson, B.:
Developing IP Multicast Networks.
Indianapolis: Cisco Press, 2000
- [Wohl01] Wohlwend, H.:
An E-factory Vision: Building On e-Diagnostics
In: Future Fab Intl. 11 (2001)
http://www.future-fab.com/document.asp?d_id=624# (23.04.2007)
- [X500-93] ISO/ITU X.500(1988) specifications:
X.500 Evolution: X.500, 1993
- [Xu03] Xu, Z. u.a.:
RITA: Receiver Initiated Just-in-Time Tree Adoption for Rich
Media Distribution.
In: ACM: Proceedings of the 13th International Workshop on
Network and operating Systems Support for Digital Audio and
Video (NOSSDAV03), June 1-3, 2003, Monterey, CA, USA.
New York: ACM Press, 2003, S. 50-59
- [Yeong95] Yeong, W.; Howes T.; Kille S.:
Lightweight Directory Access Protocol
RFC 1777. IETF OSI-DS, March 1995
- [Zhuang01] Zhuang, S.Q. u.a.:
Bayeux: an architecture for scalable and fault-tolerant wide-area
data dissemination.
In: ACM: Proceedings of the 11th International Workshop on Net-
work and Operating Systems Support for Digital Audio and Video

NOSSDAV 2001, June 25-26, 2001, Port Jefferson, New York,
USA.

New York: ACM Press, 2001, S. 11-20

10 Anhänge

10.1 Anhang A - Namensdienste

Beispiele für einen Namensdienst sind:

- Java Naming and Directory Interface (JNDI) [JNDI02],
- Remote Method Invocation (RMI) Registry [RMI06],
- Internet Domain Name System (DNS) [Mock87],
- Global Name Service (GNS) [Weber00],
- X.500 Directory Service [X500-93, Weber00],
- Lightweight Access Protocol (LDAP) [Yeong95, Weber00] und
- Interoperable Naming Service (INS) [INS99].

10.2 Anhang B - Nachrichtenorientierte Verteilungsplattformen

Innerhalb dieser Arbeit wurden die folgenden Nachrichtenorientierten Verteilungsplattformen untersucht:

- 4Tier OpenMOM²⁷
- Actional Westbridge XML Message Server²⁸
- Apache ActiveMQ²⁹
- Apache Servicemix³⁰
- Apertus MQ Edition³¹
- Arjuna Arjuna Messaging³²
- BEA messageQ³³

²⁷ <http://www.4tier.com/products/products.html> (17.07.2007)

²⁸ <http://www.actional.com/products/> (17.07.2007)

²⁹ <http://activemq.apache.org/> (17.07.2007)

³⁰ <http://incubator.apache.org/servicemix/home.html> (17.07.2007)

³¹ <http://www.apertus.com/> (17.07.2007)

³² <http://www.arjuna.com/products/index.html> (17.07.2007)

³³ <http://e-docs.bea.com/tuxedo/msgq/> (17.07.2007)

- BEA Systems Tuxedo³⁴
- Bull Flowbus³⁵
- Cape Technologies EventBroker³⁶
- CommerceQuest Architecture and Migration Planning³⁷
- CommerceQuest enableNet Data Integrator³⁸
- Constellar Hub³⁹
- Coridan Manta Ray⁴⁰
- Envoy Technologies Envoy MQ⁴¹
- Fiorano MQ 5 / MQ Server Components⁴²
- Fiorano Tifosi⁴³
- GXS InterLinx⁴⁴
- HP message service⁴⁵
- IBM WebSphere MQ⁴⁶
- IIT Software SwiftMQ⁴⁷
- Information Builders EDA / Messaging⁴⁸
- IONA Artix⁴⁹
- iPlanet Message Queues⁵⁰
- Itemfield ContentMaster⁵¹

³⁴ <http://dk.bea.com/products/tux/> (17.07.2007)

³⁵ <http://www.bull.com/servers/gcos8/products/interop8/flowbus.htm> (17.07.2007)

³⁶ <http://www.capetechnologies.com/index.html> (17.07.2007)

³⁷ http://www.metastorm.com/news/2005/060705_c.asp (17.07.2007)

³⁸ <http://www.metastorm.com/> (17.07.2007)

³⁹ <http://www.datamirror.com/products/constellar/default.aspx> (17.07.2007)

⁴⁰ http://library.theserverside.com/detail/PROD/1111666577_554.html (17.07.2007)

⁴¹ <http://www4.envoytech.com/envoy/navigate.do?iid=Products&page=Products/overview.html> (17.07.2007)

⁴² <http://www.fiorano.com/products/fmq/overview.htm> (17.07.2007)

⁴³ http://www.fiorano.com/devzone/fesb_architecture.htm (17.07.2007)

⁴⁴ <http://www.gsx.com/productsInd.htm#iqms> (17.07.2007)

⁴⁵ <http://h20219.www2.hp.com/services/cache/10942-0-0-225-121.aspx> (17.07.2007)

⁴⁶ http://www-950.ibm.com/ecatalog/Detail.wss?locale=de_DE&synkey=G106020C87422R91 (17.07.2007)

⁴⁷ <http://www.swiftmq.com/index.html> (17.07.2007)

⁴⁸ http://www.informationbuilders.com/new/magazine/v9-3/eda_evol.html (17.07.2007)

⁴⁹ <http://www.iona.com/products/artix/> (17.07.2007)

⁵⁰ http://www.sun.com/software/products/message_queue/faqs.xml (17.07.2007)

⁵¹ <http://www.ixult.com/komplexes-flatfile-mapping-mit-itemfield-contentmaster-32.html> (17.07.2007)

- JBoss JBoss MQ⁵²
- Link Development Spider II⁵³
- Logica Fastwire⁵⁴
- Microsoft MSMQ (BizTalk)⁵⁵
- My-Channels Product Overview⁵⁶
- NetMotion Mobility XE⁵⁷
- Object Web JORAM⁵⁸
- Oracle Oracle Fusion Middleware / Advanced Queuing⁵⁹
- OSMQ Open Source Message Queue⁶⁰
- PCB Systems Nirvana⁶¹
- Pramati Pramati Server⁶²
- S2 Systems⁶³
- SAP SAP Netweaver⁶⁴
- SeeBeyond e*Gate⁶⁵
- Siemens –Simatic NET⁶⁶
- Softwired iBUS⁶⁷
- Sonic Software SoniqMQ⁶⁸
- Spread Concepts LLC Spread Toolkit⁶⁹

⁵² <http://www.jboss.org/wiki/Wiki.jsp?page=JBossMQ> (17.07.2007)

⁵³ <http://linkdevelopment.com/products/action.html> (17.07.2007)

⁵⁴ <http://www.logicacmg.com/> (17.07.2007)

⁵⁵ <http://www.microsoft.com/biztalk/evaluation/adapters/adapters/msmq/2004/default.mspx> (17.07.2007)

⁵⁶ <http://www.my-channels.com/products/index.html> (17.07.2007)

⁵⁷ <http://www.netmotionwireless.com/products/> (17.07.2007)

⁵⁸ <http://joram.objectweb.org/> (17.07.2007)

⁵⁹ <http://www.oracle.com/technology/products/middleware/index.html> (17.07.2007)

⁶⁰ <http://www.pojoe.org/osmq/> (17.07.2007)

⁶¹ <http://www.nirvana.com/> (17.07.2007)

⁶² http://www.pramati.com/index.jsp?id=pro_psv41 (17.07.2007)

⁶³ <http://www.s2-systems.de/> (17.07.2007)

⁶⁴ <http://www.sap.com/platform/netweaver/index.epx> (17.07.2007)

⁶⁵ <http://www.intear.com/eGate.html> (17.07.2007)

⁶⁶ http://www.automation.siemens.com/net/index_76.htm (17.07.2007)

⁶⁷ <http://www.softwired-inc.com/> (17.07.2007)

⁶⁸ <http://www.sonicsoftware.com/products/soniqmq/index.ssp> (17.07.2007)

⁶⁹ <http://www.spread.org/index.html> (17.07.2007)

- Sterling Software Connect:MQ⁷⁰
- Sun Java System Message Queue⁷¹
- SunGard Business Integration MINT Knowledge Family⁷²
- SybasedbQ⁷³
- Sybase NEONImpact⁷⁴
- Synergy Software Technologies App-Link⁷⁵
- The OpenJMS Group OpenJMS⁷⁶
- TIBCO Rendezvous⁷⁷
- TIBCO SmartSockets⁷⁸
- Verimation VCOM⁷⁹
- Vertex Interactive Netweave⁸⁰
- Vitria BusinessWare⁸¹
- webMethods Integration Platform XML Messaging⁸²
- XmlBlaster⁸³

⁷⁰ <http://www.sterlingcommerce.com/> (17.07.2007)

⁷¹ http://www.sun.com/software/products/message_queue/index.xml (17.07.2007)

⁷² <http://www.sungard.com/STEP/default.aspx?id=77> (17.07.2007)

⁷³ <http://sybase-dbq.qarchive.org/> (17.07.2007)

⁷⁴ <http://sybase-neonimpact.qarchive.org/> (17.07.2007)

⁷⁵ <http://www.synergysw.com/> (17.07.2007)

⁷⁶ <http://openjms.sourceforge.net/> (17.07.2007)

⁷⁷ <http://www.tibco.com/software/messaging/rendezvous/default.jsp> (17.07.2007)

⁷⁸ <http://www.scientific.de/tibco.html> (17.07.2007)

⁷⁹ <http://www.verimation.com/index2.html> (17.07.2007)

⁸⁰ <http://www.netweave.com/> (17.07.2007)

⁸¹ <http://www.vitria.com/Home/index.php> (17.07.2007)

⁸² <http://www.webmethods.com/About/> (17.07.2007)

⁸³ <http://www.xmlblaster.org/> (17.07.2007)