

Strukturelle Abbildung von Produktionsnetzwerken auf Multiagentensysteme

Prof. Dr. Hans-Jürgen Appelrath • Dr. Jürgen Sauer

Tammo Freese • Thorsten Teschke

Aufgrund der Globalisierung von Märkten können Unternehmen nicht mehr isoliert voneinander betrachtet werden. Sie sind vielmehr Teil räumlich verteilter Produktions- und Logistiknetzwerke, in denen neben dem eigentlichen Produktionsprozess Transport und Lagerhaltung an Bedeutung gewinnen. Dieser Beitrag stellt einen Ansatz zur Abbildung dynamischer betrieblicher Organisationsstrukturen auf ein Multiagentensystem zur integrierten Ablaufplanung vor. Desweiteren werden Anforderungen an eine Agentenplattform für dieses System identifiziert, von denen ausgehend eine Systemarchitektur auf Basis des Komponentenmodells Enterprise JavaBeans vorgeschlagen wird.

1 Einleitung

In der Vergangenheit wurde ein Unternehmen vielfach als abgeschlossene Einheit mit klar definierten Geschäftsbeziehungen zu Abnehmern und Zulieferern betrachtet. Optimierungsmaßnahmen beschränkten sich auf die Gestaltungsspielräume innerhalb des Unternehmens. Heute sind die Kompetenzen zur schnellen und wirtschaftlichen Entwicklung und Herstellung komplexer Produkte jedoch auf verschiedene Unternehmen verteilt. Diese Entwicklung sowie der Wandel der Märkte zu Käufermärkten und kürzere Produktlebenszyklen machen Optimierungen erforderlich, die über Unternehmensgrenzen hinausgehen. Mit Konzepten wie *Supply Management*, *Supply Chain Management* und schließlich *Virtuellen Unternehmen* wird versucht, neue Optimierungspotentiale durch die Betrachtung der Verflechtungen eines Unternehmens mit Zulieferern und Abnehmern zu erschließen ([SB99, Sch98]).

2 Motivation und Projektskizze

Die Globalisierung von Märkten führt zur Entstehung räumlich verteilter Produktions- und Logistiknetzwerke, in denen neben dem eigentlichen Produktionsprozess Transport und Lagerhaltung an Bedeutung gewinnen, da sie die Termintreue entscheidend beeinflussen. Daher können zentralisierte Ansätze für die Produktionsplanung eines Unternehmens mit nur einem Standort nicht direkt auf verteilte Unternehmen übertragen werden. Zudem nimmt bei zentraler Planung die Planstabilität ab, da selbst lokal auflösbare Störungen wie Maschinenausfälle den zentralen Plan ändern. Stattdessen bietet sich der Ansatz des *Multi-Site Scheduling* ([Sau98]) an, bei dem sich die Ablaufplanung über zwei Hierarchieebenen erstreckt und auf der oberen, globalen Ebene mit ungenauen, kumulierten Informationen arbeitet. Auf globaler Ebene wird für das gesamte Unternehmen auf Basis aggregierter Kapazitäten ein Grobplan erstellt, der den einzelnen Standorten auf der lokalen

Ebene vorgegeben wird. Diese Vorgaben werden in letzteren zu standortbezogenen Plänen verfeinert. Die Verteilung der Planung bewirkt eine erhöhte Planstabilität, da lokal auflösbare Abweichungen nicht mehr im globalen Planungsvorgang berücksichtigt werden müssen.

Das Projekt AMPA (*Agent-Based Multi-Site Planning and Scheduling Application Framework*) befasst sich mit der verteilten Ablaufplanung in dynamisch veränderlichen Logistiknetzwerken ([AFST99]). Ausgehend vom Ansatz des *Multi-Site Scheduling* soll der Dynamik der „Unternehmenslandschaft“ und der Individualität einzelner Betriebe durch den Einsatz von Multiagentensystemen Rechnung getragen werden, die nach [Hen98] zur adäquaten Abbildung von Ablaufplanungsproblemen geeignet sind. Zu diesem Zweck wird das *Multi-Site Scheduling* in verschiedener Hinsicht erweitert. So wird eine integrierte Betrachtung von Produktions- und Transportplanung sowie Lagerhaltung angestrebt. Darüber hinaus wird die Beschränkung auf zwei Planungsebenen aufgehoben, wodurch eine feinere Zerlegung der Planung erreicht wird. Schließlich wird die streng hierarchische Strukturierung der im *Multi-Site Scheduling* betrachteten Planungsprobleme um eine netzwerkartige Dimension ergänzt, die insbesondere zur Abbildung zwischenbetrieblicher Beziehungen geeignet ist.

Da es hinsichtlich des zur Erfüllung der Planungs- und Koordinierungsaufgaben genutzten Wissens sowie der eingesetzten Heuristiken und Strategien zu Unterschieden zwischen Agenten im Produktions-, Transport- und Lagerhaltungsbereich kommt, soll für die Entwicklung der AMPA-Agentenplattform und -architektur ein komponentenbasierter Ansatz verfolgt werden. Spezifische Agententypen können dann durch den Austausch und die Konfiguration von Softwarekomponenten erstellt werden (vgl. hierzu auch [SSS98] und [SABH97]).

3 Grundlagen

Im Rahmen der vorangegangenen inhaltlichen Einordnung des Projekts AMPA haben sich die Schwerpunktthemen „Supply Chain Management“, „Virtuelle Unternehmen“, „Software-

agenten“ und „Komponentensoftware“ abgezeichnet, die im Folgenden detaillierter erläutert werden sollen.

3.1 Supply Chain Management

Sinkende Transaktionskosten, steigende Beherrschung der Abläufe und Profit-Center-Denken führen zunehmend dazu, dass Unternehmen Teile der Wertschöpfungskette auslagern, in denen sie keine Kernkompetenzen aufweisen (*Outsourcing*). Der parallel dazu im Zuge der Globalisierung der Märkte wachsende Zwang zu kürzeren Lieferzeiten und Produktinnovationszyklen unter gleichzeitiger Steigerung der Rendite erfordert eine Intensivierung der Zusammenarbeit aller Unternehmen entlang der unternehmensübergreifenden Wertschöpfungskette (*Supply Chain*) ([SB99]).

Die Begriffe Wertschöpfungskette und Supply Chain sind insofern irreführend, als dass die an Entwicklung und Herstellung sowie Transport, Vertrieb und Verkauf eines Produkts beteiligten Unternehmen meist keine Kette, sondern ein Netzwerk bilden. Das *Supply Chain Management* koordiniert alle Aktivitäten in diesem *Logistiknetzwerk* mit dem Ziel, den Entwicklungs- und Herstellungsprozess eines Produkts sowie die Innovation von Prozessen unternehmens- und standortübergreifend zu optimieren.

Kennzeichnend für Supply Chain Management ist die strategische, langfristige Zusammenarbeit der Hersteller sowie eine geringe Anzahl von Zulieferern für ein Produkt. Grundlage solcher Kooperationen ist massiver Informationsaustausch, der Vertrauen zwischen den Partnern der Supply Chain und den langfristigen Abbau von Informationsbarrieren zwischen den Unternehmen voraussetzt.

Zu den Risiken des Supply Chain Managements gehören die Gefahr der Entstehung einseitiger Abhängigkeiten sowie der mögliche Missbrauch der Kenntnisse über Ko-Hersteller. Außerdem ist das Supply Chain Management aufgrund der langfristigen Ausrichtung nicht für kurze Kooperationszeiten geeignet ([Sch98]).

3.2 Virtuelle Unternehmen

Ein virtuelles Unternehmen (engl. *virtual enterprise*) ist ein Netzwerk von Unternehmen, das sich zur Erfüllung einer zeitlich beschränkten Aufgabe wie z.B. einem spezifischen Kundenauftrag zusammengeschlossen hat und für die Dauer der Aufgabenerfüllung als Einheit auftritt. Anders als beim *Supply Chain Management*, das strategische Beziehungen von eher langfristiger Natur betrachtet, zeichnen sich virtuelle Unternehmen insbesondere durch ihre Fähigkeit zur kurzfristigen Reorganisation aus. Ein virtuelles Unternehmen kann als kurzzeitig bestehende Supply Chain verstanden werden.

Virtuelle Unternehmen werden mit dem Ziel gebildet, um flexibel auf die Möglichkeiten eines Marktes mit hoher Dynamik zu reagieren. Dabei werden für die Auswahl von Kooperationspartnern anstelle von eher traditionellen Faktoren wie Organisationsgröße, geographischem Standort, IT-Infrastruktur, eingesetzten Technologien und implementierten Prozessen Größen wie Kosteneffektivität und Einzigartigkeit der Produkte betrachtet. Die an einem virtuellen Unternehmen beteiligten Partner teilen zur Erfüllung der Aufgabe ihr Wissen,

ihre Kompetenzen und ihre Geschäftsbeziehungen. Diese Kombination soll es ihnen ermöglichen, globale Märkte mit Produkten und Lösungen zu erreichen, die alleine nicht zu erbringen wären ([Sch98]).

3.3 Softwareagenten

Softwareagenten eignen sich zur verteilten Problemlösung und werden in vielen Systemen der VKI verwendet. Im allgemeinen Sprachgebrauch ist ein Agent jeder, der im Auftrag oder im Interesse eines anderen handelt. In der Informatik wird der Begriff des Agenten uneinheitlich definiert. Franklin und Graesser bieten in [FG96] eine Gegenüberstellung zahlreicher Definitionen.

Wooldridge definiert einen Agenten als „ein Computersystem, das in eine Umgebung eingebettet autonom Aktionen ausführt, um die ihm gestellten Aufgaben zu erfüllen“ (frei übersetzt nach [Woo99]). Eine grundlegende Eigenschaft eines Agenten ist mithin *Autonomie* (autonomy): Ein Agent arbeitet ohne direkten Eingriff von Menschen oder anderen Systemen und hat Kontrolle über sein Verhalten und seinen internen Zustand. Der Begriff des intelligenten Agenten erweitert diese Definition um die Möglichkeit zu flexiblem Handeln, wobei der Begriff der Flexibilität drei Eigenschaften umfasst:

- *Reaktivität* (reactivity): Agenten nehmen ihre Umgebung wahr und reagieren auf Änderungen, um die ihnen gestellten Aufgaben zu erfüllen.
- *Proaktivität* (pro-activeness): Die Agenten reagieren nicht nur auf Änderungen ihrer Umgebung, sondern entwickeln ein zielgerichtetes Verhalten, indem sie die Initiative ergreifen.
- *Soziale Fähigkeiten* (social ability): Agenten interagieren mit anderen Agenten (und möglicherweise Menschen), indem sie untereinander Informationen auf Basis einer Kommunikationssprache austauschen. Soziale Fähigkeiten umfassen darüber hinaus komplexere Verhaltensweisen, die auf Konversationsprotokollen basieren.

Diese Definition soll auch für die in AMPA zu entwickelnden Planungsagenten übernommen werden: Jeder Agent soll die Fähigkeiten haben, seine eigenen Aktivitäten zu planen (Autonomie), seinen Plan bei Störungen zu ändern (Reaktivität) und selbständig zu optimieren (Proaktivität). Mitteilungen über Änderungen, Störungen usw. werden mittels einer für alle Agenten vereinbarten Sprache ausgetauscht (soziale Fähigkeiten).

Erweiterte Definitionen des Agentenbegriffs stellen zusätzliche Anforderungen. Oft wird ein Agent im engeren Sinne anhand von mentalen Begriffen wie Wissen, Glauben, Wünschen und Absichten charakterisiert, oder es werden sogar emotionale Agenten in Betracht gezogen. Beispiele für weitere Eigenschaften sind ([WJ95]):

- *Mobilität* (mobility): Mobile Agenten können sich in einem elektronischen Netzwerk bewegen.
- *Ehrlichkeit* (veracity): Ein ehrlicher Agent gibt nicht wissentlich falsche Informationen (z.B. über die Umwelt oder seinen internen Status) weiter.

- *Wohllollen* (benevolence): Wohlwollende Agenten haben keine sich widersprechenden Ziele, und sie versuchen grundsätzlich das zu tun, wozu sie aufgefordert werden.
- *Vernunft* (rationality): Rationale Agenten versuchen, ihre Ziele zu erreichen, und handeln nicht wissentlich entgegen ihren Zielen.

Diese Aspekte werden in verschiedenen Ansätzen der VKI untersucht.

3.4 Komponentensoftware

Hinter dem Begriff der Komponentensoftware („componentware“) steht eine Idee, die in allen reifen Ingenieursdisziplinen Anwendung findet: Softwarekomponenten (nachfolgend nur noch Komponenten genannt) sollen ingenieurmäßig entworfen, erstellt und eingesetzt werden. Neben dem Aspekt der Wiederverwendung besteht bei der Komponentensoftware ein wichtiger Anspruch in der Integrationsfähigkeit. Anwendungssysteme sollen in komfortabler Weise durch das Zusammenfügen von Komponenten erstellt werden können („Plug and Play“) ([Fra99]). Szyperski definiert eine Komponente als „[...] eine Kompositionseinheit, die nur vertraglich vereinbarte Schnittstellen und explizite Kontextabhängigkeiten aufweist. Eine Komponente kann unabhängig eingesetzt werden und unterliegt der Komposition durch dritte [Parteien]“ (frei übersetzt nach [Szy98]). Dabei betont Szyperski, dass Komponenten als Konsequenz dieser Definition binäre Einheiten sind, die ohne Modifikation zusammengefügt werden.

Eine Voraussetzung für die Komposition komplexer Systeme aus einzelnen Komponenten ist die Standardisierung der angebotenen Schnittstellen durch sogenannte *Komponentenmodelle*. Neben der Definition eines Modells für die Kopplung einzelner Komponenten werden abhängig vom jeweiligen Komponentenmodell weitere Schnittstellen spezifiziert.

4 Abbildung betrieblicher Organisationsstrukturen auf Softwareagenten

Die Grundelemente betrieblicher Organisationen sind *Stellen* sowie *Beziehungen* zwischen diesen. Wichtige Dimensionen betrieblicher Organisationssysteme sind Spezialisierung, Koordination sowie das Leitungssystem. Der Aspekt der *Spezialisierung* betrifft die Art der Arbeitsteilung, bei der Teilaufgaben unterschiedlicher Art von verschiedenen Organisationseinheiten wahrgenommen werden. Aus der durch die Spezialisierung erreichten Arbeitsteilung resultiert die Notwendigkeit der *Koordination* der Aktivitäten. Diese kann durch den Einsatz von Hierarchien vereinfacht werden. Das *Leitungssystem* spezifiziert Verantwortlichkeiten, Entscheidungs- und Weisungsbefugnisse, die eine übergeordnete Stelle gegenüber einer untergeordneten Stelle hat. Hinsichtlich der Strukturierung der Stellen im Leitungssystem können zwei idealtypische Grundformen unterschieden werden. Das *Einliniensystem* beruht auf dem Prinzip der Einheit der Auftragserteilung und organisiert die Stellen in einer Baumstruktur. Das *Mehrliniensystem* soll helfen, das Prinzip des kürzesten Weges bei abteilungsübergreifenden Koordinierungsproblemen zu

realisieren und organisiert die Stellen in einer Graphstruktur. Dabei kann eine Stelle mehreren Stellen innerhalb der Hierarchie unterstellt werden, wodurch sich allerdings die Gefahr von Kompetenzstreitigkeiten und unklaren Verantwortungsbeziehungen ergibt ([Sch93]).

4.1 Organisationsmodell

Das in AMPA vorgeschlagene Organisationsmodell erweitert den in Abschnitt 2 skizzierten Ansatz des Multi-Site Scheduling in zweierlei Hinsicht: Zum einen wird die betrachtete Hierarchie um zusätzliche Stufen erweitert, zum anderen wird die hierarchische, innerbetriebliche Betrachtungsweise um eine netzwerkartige, zwischenbetriebliche Dimension ergänzt.

Eine für innerbetriebliche Leitungssysteme übliche Form sind hierarchische Strukturen, da sie sich für die Definition von Entscheidungsbefugnissen, Weisungskompetenzen, Aufsichtspflichten und Kontrollrechten eignen. Dieser Ansatz soll auch im Rahmen von AMPA verfolgt werden. Dazu werden Stellen nach ressourcenorientierten Gesichtspunkten definiert und in eine Hierarchie eingeordnet. Mögliche Stellen sind beispielsweise das gesamte Unternehmen, Standorte, Werkstätten, Lager, Transportfahrzeuge, Ressourcengruppen oder Maschinen. Eine Stelle wird jeweils durch einen Planungsagenten repräsentiert.

Komplexe Organisationsformen sind allerdings nicht ausschließlich hierarchisch organisiert. Dies gilt in besonderem Maße für rechtlich und wirtschaftlich selbständige Unternehmen, da Beziehungen zwischen diesen weder Entscheidungsbefugnisse noch Weisungskompetenzen definieren, sondern vielmehr den Aspekt der Koordinierung zwischen den Leitungssystemen darstellen. Neben der hierarchischen, eher statischen Dimension des innerbetrieblichen Leitungssystems wird daher in AMPA auch die netzwerkartige, tendenziell dynamische Dimension der koordinierenden, logistischen Beziehungen auf allen Hierarchieebenen durch einen speziellen Beziehungstyp berücksichtigt. In AMPA werden Organisationen folglich durch die Überlagerung hierarchischer und netzwerkartiger Strukturen repräsentiert. Abbildung 1 illustriert das in AMPA verwendete Organisationsmodell. Dabei sind interne Organisationseinheiten eines Unternehmens durch dunkel gefärbte Knoten dargestellt. Externe Organisationseinheiten sind als helle Knoten abgebildet.

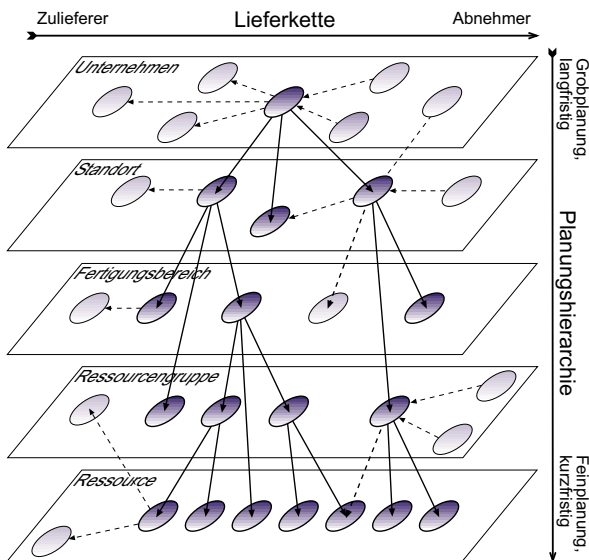


Abbildung 1: Organisationsmodell

Auf einer formelleren Ebene kann der im Rahmen des Projektes AMPA verfolgte Ansatz als Kombination der Idealtypen des Ein- und Mehrliniensystems bezeichnet werden. Damit wird das Ziel verfolgt, klare Verantwortlichkeiten einerseits und verbesserte Koordinierungsmöglichkeiten durch kürzere Wege andererseits zu erhalten. Um dieses Ziel unter Vermeidung der zuvor aufgeführten Nachteile zu erreichen, werden zwei unterschiedliche Beziehungstypen betrachtet. Beide Typen sind gerichtet und definieren eine potentielle Nutzung einer Stelle durch eine andere, wobei im Folgenden erstere als *Zulieferer* und letztere als *Abnehmer* bezeichnet wird. Die Nutzung besteht darin, dass der Zulieferer Leistungen für den Abnehmer erbringt.

Disziplinarische Unterstellungsverhältnisse werden als Bestandteil des Leitungssystems genutzt, um die innerbetriebliche hierarchische Dimension des Organisationsmodells abzubilden. Sie spezifizieren über die Nutzungsbeziehung hinaus die alleinige Verantwortlichkeit einer übergeordneten Stelle für die ihr disziplinarisch untergeordneten Stellen. Um die hierfür notwendige Eindeutigkeit zu erreichen, kann eine Stelle maximal einer anderen Stelle untergeordnet sein, d.h. die Struktur der disziplinarischen Unterstellungsverhältnisse entspricht dem Einliniensystem. Die Verantwortlichkeit einer Stelle für eine andere äußert sich in dem von AMPA adressierten Problembereich dadurch, dass Bestellanforderungen einer Stelle an die disziplinarisch vorgesetzte Stelle gemeldet werden können, wenn die Anforderungen durch keine Leistung der dieser Stelle untergeordneten Stellen erfüllt werden können. Darüber hinaus spezifiziert ein disziplinarisches Unterstellungsverhältnis, dass der übergeordneten Stelle in angemessenem Umfang Zugriff auf Informationen über den Zustand (z.B. die Auslastung) der untergeordneten Stelle gewährt werden muss. Diese Information kann den Entscheidungsfindungsprozess der übergeordneten Stelle positiv beeinflussen. Disziplinarische Beziehungen sind in Abbildung 1 durch durchgezogene Pfeile dargestellt.

Funktionale Beziehungen bilden die netzwerkartige Dimension des Organisationsmodells ab und dienen der Koordinierung zwischen Organisationseinheiten, die nicht im Leitungssystem verbunden sind. Eine funktionale Beziehung definiert eine reine Nutzungsbeziehung zwischen zwei Stellen, d.h. der

Zulieferer darf weder Anforderungen an den Abnehmer stellen noch erwächst dem Abnehmer in irgendeiner Form Verantwortung für den Zulieferer. Darüber hinaus ist der Zulieferer nicht verpflichtet, dem Abnehmer Informationen über seinen Zustand preiszugeben. Eine Stelle kann neben einer disziplinarischen Unterstellung beliebig viele funktionale Beziehungen pflegen, so dass das System der funktionalen Beziehungen dem Mehrliniensystem entspricht. Funktionale Beziehungen werden in Abbildung 1 durch gestrichelte Pfeile repräsentiert.

Einen zu dem hier vorgestellten Organisationsmodell vergleichbaren Ansatz verfolgen Swaminathan et al. in [SSS98], die die strukturellen Elemente einer Supply Chain wie z.B. Produktions- und Transporteinheiten durch Agenten repräsentieren. Allerdings werden dort keine hierarchischen Beziehungen berücksichtigt, d.h. eine Supply Chain wird als flaches Netzwerk abgebildet. Weitere agentenbasierte Ansätze zur Organisationsmodellierung legen oft einen Schwerpunkt auf die Erstellung von Ontologien zur Unterstützung der Abbildung von Abläufen in Unternehmen [FG98]. Das Enterprise Projekt [Sta96] setzt ebenfalls Agenten ein, die jedoch lediglich Werkzeuge oder Menschen, die Tätigkeiten verrichten können, repräsentieren. Diese werden in ein System, das die Ausführung von Workflows steuert, integriert. Im TOVE Projekt [BF94] werden "klassische" Funktionen der Produktionsplanung und -steuerung durch Agenten repräsentiert, die durch sogenannte Informationsagenten verknüpft werden.

4.2 Abbildung auf Softwareagenten

Im Projekt AMPA wird ein Unternehmen durch ein System von Agenten dargestellt. Sie gestatten die direkte Nachbildung vorhandener Strukturen und Kommunikationswege sowie die Beibehaltung vorhandener Ablaufplanungssysteme. Als einzelne Schritte zur Darstellung sind erforderlich:

- 1) *Identifikation der Agenten:* Unternehmensinterne Organisationseinheiten (Stellen), für die in AMPA geplant werden soll, werden jeweils durch einen Agenten repräsentiert. Vorhandene Systeme zur Ablaufplanung einer oder mehrerer Organisationseinheiten, die bei der Einführung der Ablaufplanung mittels AMPA beibehalten werden sollen, sind durch einen Agenten zu kapseln, der als Schnittstelle zu AMPA dient. Die Ablösung bestehender Planungssysteme ist somit nicht erforderlich, womit der offenen, dynamischen Planungsumgebung Rechnung getragen wird.
- 2) *Ergänzung disziplinarischer Beziehungen:* Zwischen den durch die Agenten repräsentierten Stellen bestehen in einem Unternehmen disziplinarische Unterstellungsverhältnisse. Die Agenten werden dann durch diese disziplinarischen Unterstellungsverhältnisse in einer Baumstruktur organisiert. Ergibt sich keine Baumstruktur, müssen Konflikte geeignet aufgelöst werden.
- 3) *Ergänzung innerbetrieblicher funktionaler Beziehungen:* Neben disziplinarischen Beziehungen liegen in einem Unternehmen auch funktionale Beziehungen vor, die auf das System von Planungsagenten zu übertragen sind.

Die zuvor aufgeführten Aktivitäten dienen der Repräsentation der internen Strukturen eines Unternehmens. Die Integration

des Unternehmens in seine Umgebung wird durch den folgenden Schritt erreicht.

- 4) *Einbindung von Zulieferern und Abnehmern:* Um ein Unternehmen in das logistische Netzwerk zu integrieren, sind die direkten Zulieferer und Abnehmer des Unternehmens in das System einzubinden. Setzen diese AMPA-Agenten ein, müssen dazu die entsprechenden Agenten identifiziert werden, ansonsten sind für die Kommunikation entsprechende Schnittstellenagenten zu erstellen. Sollen große Gruppen gleichartiger Abnehmer oder Zulieferer integriert werden, können diese alternativ durch einen gemeinsamen Schnittstellenagenten repräsentiert werden. Die Agenten der Zulieferer und Abnehmer werden schließlich mit den Agenten, die von ihnen Dienste erfragen bzw. ihnen Dienste anbieten, funktional in Beziehung gesetzt.

5 Entwurf einer Agentenplattform

Eine Voraussetzung für den Einsatz eines Multiagentensystems ist die Existenz einer Plattform, die eine Umgebung für die jeweilig genutzten Agenten darstellt.

5.1 Anforderungen

Aus der durch das Projekt AMPA adressierten Problemstellung ergeben sich eine Anzahl von Anforderungen an eine Agentenplattform. Die im Hinblick auf das vorgestellte Organisationsmodell wichtigsten Anforderungen werden nachfolgend erläutert:

- *Verteilung:* Im Rahmen der in AMPA betrachteten Problemstellung kann zwischen einer *lokalen* Verteilung, bei der mehrere Agenten auf einem Server laufen, und einer *globalen* Verteilung, bei der die Agenten auf verschiedene Server verteilt eingesetzt werden, unterschieden werden. So können die Agenten eines Unternehmens beispielsweise auf einem Server laufen, die an einem virtuellen Unternehmen beteiligten oder innerhalb einer Supply Chain verbundenen Unternehmen werden im allgemeinen jedoch jeweils eigene Server betreiben.
- *Kommunikation:* Die Forderung nach unterschiedlichen Formen der Verteilung beinhaltet die Notwendigkeit zur Kommunikation sowohl zwischen lokal als auch global verteilten Agenten. Die Kommunikationsschnittstelle einer Agentenplattform sollte diesen Unterschied kapseln, so dass es für einen Agenten transparent ist, ob sich sein Kommunikationspartner auf demselben Server oder einem anderen, entfernten Server befindet.
- *Plattformunabhängigkeit:* Der verteilte Einsatz von Agenten in einer offenen, heterogenen Umgebung erfordert entweder die Entwicklung spezifischer Agenten für jede eingesetzte Plattform oder aber eine Agentenplattform, die die Spezifika einer Umgebung kapselt und eine einheitliche Umgebung für die Agenten bietet. Die erste Variante erscheint gerade im Umfeld virtueller Unternehmen, die ein hohes Maß an Heterogenität bzgl. der eingesetzten Informations- und Kommunikationstechnologien aufweisen, nicht gangbar. Stattdessen bietet sich an, auf Basis der Programmiersprache Java eine Agentenplattform zu

entwickeln, die weitgehend unabhängig von der vorhandenen I+K-Infrastruktur ist.

- *Sicherheit:* Im Bereich der Multiagentensysteme, bei denen Softwareagenten reale Systeme teil-autonom repräsentieren, kommt der Frage nach Sicherheit besondere Bedeutung zu. Eine Agentenplattform muss geeignete Mechanismen anbieten, um einen Agenten vor missbräuchlichem Zugriff, z.B. durch Wettbewerber auf globalen Märkten, zu schützen. Um Absicherung gegen unbefugte Zugriffe einerseits und vollständige Kontrolle durch spezielle Benutzer andererseits zu ermöglichen, ist ein Berechtigungskonzept zu realisieren, das Rollen mit unterschiedlichen Zugriffsrechten unterstützt.

Weitere Anforderungen, die nicht in direktem Zusammenhang mit der Abbildung betrieblicher Organisationsstrukturen stehen, werden nachfolgend skizziert:

- *Persistenz:* Ein Planungsagent benötigt den Zugriff auf persistente Informationen wie z.B. den Produktionsplan oder das Produktmodell. Um jedoch obige Forderung nach Plattformunabhängigkeit erfüllen zu können, muss die Agentenplattform die Entkopplung der Anwendungsebene (den Agenten) von der Persistenzebene ermöglichen.
- *Benutzungsschnittstellen:* Eine für die Akzeptanz eines Multiagentensystems durch menschliche Benutzer oftmals wichtige Eigenschaft ist die Möglichkeit, einzelne Agenten zu überwachen und steuernd eingreifen zu können. Aus diesem Grund sollte eine Agentenplattform die Möglichkeit bieten, entsprechende graphische Benutzungsoberflächen mit den Agenten zu verbinden.
- *Transaktionen:* Komplexe Planungen, die eine Anzahl von aufeinanderfolgenden Verhandlungen mit anderen Agenten entlang der Lieferkette erfordern, unterliegen der Gefahr, das sie nur unvollständig ausgeführt werden können und somit zu inkonsistenten Plänen führen. Abhilfe kann hier das Konzept der Transaktionen schaffen.
- *Konfigurierbarkeit:* Die Anforderungen bzgl. Persistenz und Sicherheit machen z.B. den Austausch von Datenbankmanagementsystemen oder ERP-Softwaresystemen sowie die Einrichtung von Benutzergruppen und die Vergabe von Rechten an diese Gruppen notwendig. Darüber hinaus sollten auch die eingesetzten Agenten konfigurierbar sein.

Eine Grobarchitektur zur Umsetzung dieser Anforderungen ist in Abbildung 2 dargestellt. Eine herausragende Rolle nimmt dabei die Kommunikationsschnittstelle ein. Sie sorgt einerseits für die Vermittlung von Nachrichten zwischen Agenten und gewährleistet andererseits die Einhaltung des Sicherheitskonzepts.

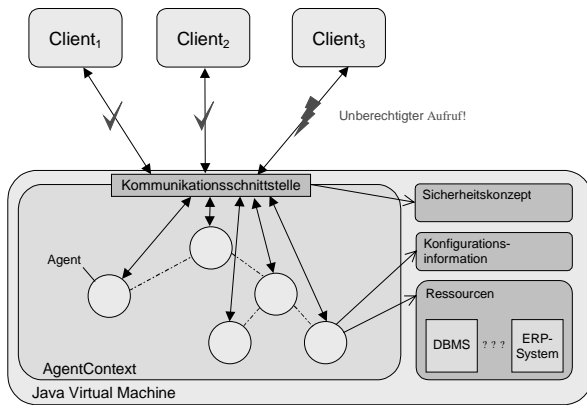


Abbildung 2: Grobarchitektur

Die Anforderung hinsichtlich der Konfigurierbarkeit soll im Rahmen von AMPA durch den Einsatz von Methoden der komponentenbasierten Softwareentwicklung (vgl. Abschnitt 3.4) erreicht werden. Dadurch soll es ermöglicht werden, ausgehend von einem generischen Agententypen spezialisierte Agententypen z.B. für Produktion, Lagerhaltung und Transport durch den Austausch und die Konfiguration einzelner Komponenten zu entwickeln (vgl. hierzu auch [SSS98]).

5.2 Enterprise JavaBeans

Aktuell wird der Ansatz verfolgt, das Komponentenmodell *Enterprise JavaBeans (EJB)* von Sun Microsystems ([MH99]) als Grundlage für die Realisierung der AMPA-Agentenplattform einzusetzen. EJB ist auf die Entwicklung mehrschichtiger, verteilter Anwendungen ausgerichtet und bietet für die wichtigsten der in Abschnitt 5.1 aufgeführten Anforderungen geeignete Konzepte.

Die durch EJB spezifizierte Architektur umfasst drei Bestandteile. Der *Server* stellt eine Umgebung für die Ausführung von Anwendungen bereit. Typische Aufgaben eines Servers umfassen die Bereitstellung von Anbindungen an Datenbankmanagementsysteme, das Transaktionsmanagement und allgemeine Dienste wie z.B. die Lastverteilung zwischen mehreren Servern. Innerhalb eines Servers verwalten *Container* die Komponenten des EJB-Komponentenmodells (die *Enterprise Beans*). Sie stellen eine Schnittstelle für die systemunabhängige Programmierung der Clients und der Enterprise Beans zur Verfügung, die in Form von Verträgen festgelegt wird. Der *Client Contract* ist dabei der Vertrag zwischen Client und Container, der den Zugriff eines Clients auf die in einem Container eingesetzten Komponenten spezifiziert. Für jede Komponente stellt der Container ein sogenanntes *Home-Interface* zur Verfügung, das Methoden zur Erzeugung, zum Auffinden und zum Löschen von Geschäftsobjekten anbietet. Diese lassen sich über *Remote-Interfaces* ansprechen, die vom Container implementiert werden und anwendungsspezifische Methoden der Komponente bereitstellen. Der *Component Contract* spezifiziert die Schnittstelle zwischen einem Container und einer darin enthaltenen Enterprise Bean. Er umfasst Vereinbarungen über den Lebenszyklus einer Enterprise Bean, die Verwaltung von Transaktionen und Sicherheit, die Bereitstellung von Persistenz- und Namensdiensten und über die zu implementierenden Schnittstellen. Es werden zwei Typen von Enterprise Beans unterschieden. *Session Beans* sind transient,

d.h. sie repräsentieren keine persistent zu haltende Daten. Sie existieren nur für die Dauer einer Sitzung und sind nur für einen Client verfügbar. *Entity Beans* repräsentieren persistente Objekte, die in einer Datenbank gehalten werden und auf die mehrere Clients gemeinsam zugreifen können.

Das Sicherheitsmanagement von EJB basiert auf der Entkopplung der Sicherheitsinfrastruktur von dem anwendungsspezifischen Sicherheitskonzept. Diese Architektur befreit den Entwickler einer Enterprise Bean von dem Zwang, Sicherheitskonzepte und -mechanismen in der Komponente zu implementieren. Erst bei der Erstellung einer Anwendung definiert der Anwendungsentwickler Rollen (*security roles*), die jeweils zur Ausführung bestimmter Methoden berechtigen. Aufgabe des Administrators ist es, den Rollen innerhalb der operativen Einsatzumgebung existierende Auftraggeber (*principals*) zuzuordnen. Ein Client ist nur berechtigt, eine Methode aufzurufen, wenn der mit dem Aufruf assoziierte Auftraggeber zumindest eine Rolle hat, die eine Berechtigung zum Aufruf der Methode umfasst.

Fragen der Konfiguration des Containers bzw. der komponentenbasierten Anwendungen werden im Rahmen von EJB durch sogenannte *Deployment Descriptors (DD)* adressiert. Dies sind XML-Dokumente, die auf einer durch EJB spezifizierten *Document Type Definition* basieren und zwischen den an der Softwareentwicklung beteiligten Partnern ausgetauscht und entsprechend des Anwendungsentwicklungs- und -einsatzprozesses fortgeschrieben werden. DDs beinhalten sowohl strukturelle Informationen einzelner Enterprise Beans als auch Konstruktionsinformationen größerer Softwaresysteme, die aus mehreren Beans bestehen.

5.3 Architektorentwurf auf Basis von EJB

Der Architektorentwurf der AMPA-Agentenplattform weist eine geschachtelte Struktur auf (vgl. Abbildung 3). Die äußerste Schale bildet dabei der *AgentServer*, der auf der Java Virtual Machine aufsetzt und die für die Architektur relevanten Prozesse umschließt. Zu diesen Prozessen gehören einerseits *AgentContexts*, die ähnlich dem Container in der EJB-Spezifikation die Anwendungslogik beherbergen, und andererseits das Java Naming and Directory Interface (JNDI). Darüber hinaus kapselt der AgentServer Ressourcen wie z.B. Datenbankmanagementsysteme, ERP-Software oder Drucker und stellt diese dem AgentContext bzw. der darin in Form von Komponenten enthaltenen Anwendungslogik zur Verfügung.

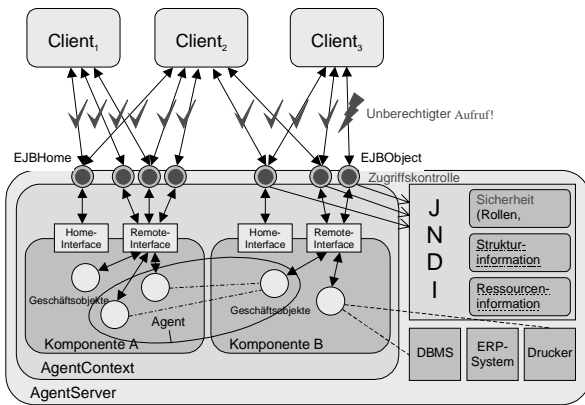


Abbildung 3: Architektur der Agentenplattform

Ein Kernelement der Systemarchitektur stellt das JNDI dar, das jede Art erforderlicher Konfigurationsinformation in standardisierter Form verwaltet. Dazu gehören u.a. Informationen über die verschiedenen Rollen des Sicherheitskonzepts, deren Berechtigungen und die Zuordnung dieser Rollen zu Organisationseinheiten, strukturelle Informationen über die Konstruktion verschiedener Agententypen durch Komposition und Parametrierung von Komponenten sowie Informationen über die durch den Server zur Verfügung gestellten Ressourcen.

Die zentrale Aufgabe des AgentContexts ist die Verwaltung von Komponenten bzw. deren Geschäftsobjekten. Die Anforderungen eines AgentContexts an die von ihm verwalteten Komponenten orientieren sich an Client Contract und Component Contract der EJB-Spezifikation. Das heißt insbesondere, dass jede AMPA-Komponente sowohl ein Home- als auch ein Remote-Interface hat. Die AMPA-Agentenplattform berücksichtigt sowohl Entity- als auch Session-Beans. Erstere können z.B. für Kundenaufträge, das Produktmodell und die geplanten Aktivitäten, letztere beispielsweise für Verhandlungssequenzen, Planungsalgorithmen und Strategien genutzt werden.

Die Implementierung der Agentenplattform wirft zwei Fragestellungen auf, bei denen entschieden werden muss, ob die EJB-Spezifikation verlassen wird. Zum einen muss der Spezifikation entsprechend jegliche Kommunikation zwischen Komponenten die Sicherheitsüberprüfungen durchlaufen. Eine performantere Alternative besteht darin, zumindest für die Komponenten eines Agenten die Sicherheitsüberprüfung durch die Verwendung direkter Referenzen zu umgehen. Zum anderen verbietet die Spezifikation, dass eine Enterprise Bean Threads erzeugt. Somit kann proaktives Handeln eines Agenten nicht direkt implementiert werden.

6 Ausblick

Das in diesem Beitrag vorgestellte Multiagentensystem zur Unterstützung der Ablaufplanung in Produktionsnetzwerken befindet sich derzeit noch im Entwurfsstadium. Zur Erstellung eines Prototyps muss eine Spezifikation der Verhandlungsprotokolle erfolgen, die zwischen den Agenten verwendet werden. Dabei sind Verhandlungen für funktionale und disziplinarische Beziehungen zu unterscheiden.

Eine häufige Kritik an Multiagentensystemen betrifft die mangelnde Performanz aufgrund zu intensiver Kommunikation

zwischen den Agenten. Das im Abschnitt 4.1 vorgestellte Organisationsmodell adressiert diese Problematik durch die Einführung expliziter Kommunikationsbeziehungen. Die Erstellung eines detaillierten verteilten Plans über alle Hierarchiestufen der Organisationsstruktur hinweg involviert dennoch eine Vielzahl von Agenten und bewirkt somit hohen initialen Einplanungsaufwand. Aufgrund der Änderungsanfälligkeit derartiger Feinplanungen entsteht zudem noch beträchtlicher reaktiver Planungsaufwand. Da langfristige Planungen oft nicht detailliert vorliegen müssen, bietet sich die Erstellung von Grobplänen an, die in Abhängigkeit von Kriterien wie dem Ausführungszeitpunkt und der Wichtigkeit der Aufträge auf unteren Planungsebenen verfeinert werden. Die Erstellung geeigneter Grobpläne erfordert Informationen über die Auslastung der Agenten auf untergeordneten Ebenen, deren adäquate Aggregation Gegenstand zukünftiger Arbeiten ist. Als Grundlage sollen dabei die Ansätze von Sycara et al. [SRSF91b] sowie Neiman et al. [NHLS94] dienen, die aus dem Bereich des verteilten Problemlösens stammen und Maße für die Nachfrage sowie die Verfügbarkeit von Ressourcen vorschlagen.

7 Literatur

- [AFST99] Appelrath, H.-J., Freese, T., Sauer, J. und Teschke, T.: *Konzept eines Multiagentensystems für die verteilte Ablaufplanung*, *Proceedings des Workshops "Agententechnologie" auf der KI '99*, Bonn, 1999.
- [BF94] Barbuceanu, M. und Fox, M. S.: *The Information Agent: An Infrastructure Agent Supporting Collaborative Enterprise Architectures*, *3rd Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, Morgantown, WV, 1994.
- [FG98] Fox, M. S. und Gruninger, M.: *Enterprise Modeling*. *AI Magazine*, 19 (3, Fall 1998), 109 - 121, 1998.
- [Fra99] Frank, U.: *Componentware - Software-technische Konzepte und Perspektiven für die Gestaltung betrieblicher Informationssysteme*. *Information Management & Consulting*, 14 (2), 11-18, 1999.
- [FG96] Franklin, S. und Graesser, A.: *Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents*, *Third International Workshop on Agent Theories, Architectures, and Languages*. Springer-Verlag, 1996.
- [Hen98] Henseler, H.: *Aktive Ablaufplanung mit Multi-Agenten*. Dissertation, Carl von Ossietzky Universität Oldenburg, Oldenburg, 1998.
- [MH99] Matena, V. und Hapner, M.: *Enterprise JavaBeans Specification*, Sun Microsystems, 1999.
- [NHLS94] Neiman, D. E., Hildum, D. W., Lesser, V. R. und Sandholm, T. W.: *Exploiting Meta-Level Information in a Distributed Scheduling System*, <http://janus.inf.ufsc.br:1082/users/a/angel/publica>.

htm, University of Massachusetts, Computer Science Department, Amherst, 1994.

- [Sau98] Sauer, J.: *A Multi-Site Scheduling System*. AAAI's Special Interest Group in Manufacturing Workshop on Artificial Intelligence and Manufacturing: State of the Art and State of Practice, 1998.
- [SABH97] Sauer, J., Appelrath, H.-J., Bruns, R. und Henseler, H.: *Design-Unterstützung für Ablaufplanungssysteme*. KI-Künstliche Intelligenz, 2/97, 37-42, 1997.
- [SB99] Scheer, A.-W. und Borowsky, R.: *Supply Chain Management: Die Antwort auf neue Logistikanforderungen*, LM'99 - Intelligente I+K Technologien, Bremen. Springer Verlag, 1999.
- [Sch93] Schierenbeck, H.: *Grundzüge der Betriebswirtschaftslehre*. Oldenbourg Verlag GmbH, München, 11. Auflage, 1993.
- [Sch98] Schönsleben, P.: *Integrales Logistikmanagement: Planung und Steuerung von umfassenden Geschäftsprozessen*. Springer, Berlin, Heidelberg, 1998.
- [Sta96] Stader, J.: *Results of the Enterprise Projekt*, Artificial Intelligence Applications Institute, Edinburgh, 1996.
- [SSS98] Swaminathan, J. M., Smith, S. F. und Sadeh, N. M.: *Modeling Supply Chain Dynamics: A Multiagent Approach*. *Decision Sciences Journal*, 29 (3), 607-632, 1998.
- [SRSF91b] Sycara, K. P., Roth, S. F., Sadeh, N. und Fox, M. S.: *Resource Allocation in Distributed Factory Scheduling*. *IEEE Expert*, 6 (1), 29-40, 1991.
- [Szy98] Szyperski, C.: *Component Software - Beyond Object-Oriented Programming*. Addison-Wesley Verlag, 1998.
- [Woo99] Wooldridge, M.: *Intelligent Agents*, In: G. Weiss (ed.): *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, 27-77. The MIT Press, 1999.
- [WJ95] Wooldridge, M. und Jennings, N. R.: *Intelligent Agents: Theory and Practice*. *Knowledge Engineering Review*, 10 (2), 1995.