Technik

Robert Neuwirth

Konzeption und Realisierung einer Middleware zur Unterstützung der Interoperabilität verteilter Software-Komponenten in der Automation

Masterarbeit

Diplom.de

Bibliografische Information der Deutschen Nationalbibliothek:

Bibliografische Information der Deutschen Nationalbibliothek: Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de/ abrufbar.

ISBN: 9783832495312

Robert Neuwirth

Konzeption und Realisierung einer Middleware zur Unterstützung der Interoperabilität verteilter Software-Komponenten in der Automation

Diplom.de

Robert Neuwirth

Konzeption und Realisierung einer Middleware zur Unterstützung der Interoperabilität verteilter Software-Komponenten in der Automation

Schwerpunkt Anforderungsanalyse

MA-Thesis / Master
Fachhochschule Reutlingen
Fachbereich Mechatronik
Januar 2005

ID 9531
Neuwirth, Robert: Konzeption und Realisierung einer Middleware zur Unterstützung der Interoperabilität verteilter Software-Komponenten in der Automation - Schwerpunkt Anforderungsanalyse
Druck Diplomica GmbH, Hamburg, 2006
Zugl.: Fachhochschule Reutlingen, MA-Thesis / Master, 2005

Diplomica GmbH
http://www.diplom.de, Hamburg 2006
Printed in Germany

Inhalt

1 Einleitung

1.1 Motivation

Die Verbreitung von Computern nimmt besonders außerhalb der klassischen Einsatzgebiete als Zentralrechner oder PC enorm zu (so sind heutzutage schon Kaffee-Maschinen mit eigenen Betriebssystemen ausgerüstet). Durch diese Verbreitung stellt sich die Forderung nach Interkommunikation und Interoperabilität. Auch die erweiterten Möglichkeiten und Funktionen einer solchen Zusammenarbeit erscheinen verlockend. Jedoch ist bisher relativ wenig Software auf dem Markt zu finden, die speziell die Interkommunikation in einer heterogenen Produktlandschaft unterstützt. Dieses Fehlen einer (optimalerweise standardisierten) Unterstützung ist besonders im Bereich von Projekten in der Automation zu beobachten. So schreibt Humphrey:

„Ein echtes Problem stellt für Herstellungsbetriebe jeglicher Art heute immer noch die fehlende Interoperabilität zwischen den Systemen dar.“ [DHID1104]

Als Folge müssen oft und mangels fehlenden, beziehungsweise nicht alle Anforderungen abdeckenden Entwicklungshilfsmitteln, Schnittstellen zwischen einzelnen Geräten und deren Computern immer wieder neu und speziell erstellt werden. Eine Arbeit auf diesem Gebiet erscheint daher viel versprechend, nicht nur zur Lösung des Integrations-Dilemmas, sondern auch im Hinblick auf die erweiterten Möglichkeiten, die solch ein Software-System bieten könnte.

1.2 Vorgehensweise

Die vorliegende Thesis nähert sich aufgrund von beispielhaften Anwendungsfällen den Anforderungen an ein die Interoperabilität unterstützendes Software-System. In diesem Rahmen wird die Notwendigkeit der Ableitung applikationsübergreifender Funktionalitäten dargestellt. Darüber hinaus werden Vorteile sowie Nutzen des Einsatzes eines Software-Systems zur Unterstützung der Entwickler von verteilten Anwendungen, sowie Lösungsansätze zur Aufnahme von allgemein nützlichen Funktionen beschrieben. Das Software-System wird im Folgenden als Middleware bezeichnet:

„Unter Middleware versteht man Software, die zwischen der Anwendungssoftware und der Betriebssystemsoftware angesiedelt ist. Kommunikations-Middleware dient der vereinfachten und standardisierten Kommunikation der Komponenten eines verteilten Systems.“ [SWTI01]

Dem Autor ist bewusst, dass schon eine ganze Reihe von Middleware-Lösungen auf dem Markt verfügbar sind. Daher wird in Kapitel 4.2 zum Zwecke der Abgrenzung eine Untersuchung von gängigen Produkten wie CORBA, OPC und Jini durchgeführt.

1.3 Aufteilung auf zwei Thesen

Sowohl die konzipierte Middleware als auch der realisierten Teile entstanden in enger Zusammenarbeit mit Stephan Zimmer [zi05]. Als Folge ist die klare Trennung einzelner Bereiche, sowie die Herkunft zugrunde liegender Ideen (speziell bei der entwickelten Architektur) im Nachhinein oft nur schwer durchführbar. Somit sollte das Gesamtergebnis als eine Gemeinschafts-Leistung betrachtet werden. Es wurde trotzdem versucht, aufgrund der jeweils gesetzten Schwerpunkte die Arbeit in zwei sich ergänzende Thesen aufzuteilen: Die vorliegende Thesis beschreibt Anforderungen und Konzepte, die anhand von Beispielen automatisierungstechnischer Anwendungen erarbeitet werden. Die Partner-Thesis [zi05] geht auf die Umsetzung der Anforderungen in eine Middleware ein. Deren exemplarische Anwendung am Beispiel einer Interaktion zwischen (in ihrem Aufbau sehr unterschiedlichen) Robotern ist wiederum in dieser Thesis beschrieben. Bei einigen fachlichen Bereichen war eine klare Arbeitsteilung möglich, so dass neben der oben beschriebenen Aufteilung einige ausgewählte Themen der Umsetzung sich in der vorliegenden Thesis widerspiegeln, insofern sie die Arbeit des Autors betreffen. Die einzelnen Thesen sind so aufgebaut, dass sie als eigenständige Dokumente betrachtet werden können. Als Folge sind bestimmte Themen in beiden Thesen zu finden, die je nach den Schwerpunkten der jeweiligen Thesis mehr oder weniger ausführlich beschrieben sind. So ergibt sich ungeachtet der jeweiligen Eigenständigkeit das umfassende Gesamtbild der entwickelten Middleware erst durch das Studium beider Thesen.

Sollte daher der Leser einzelne Aspekte und Bereiche in vorliegendem Dokument nicht oder nicht ausführlich genug beschrieben wieder finden wird sich zur Vervollständigung ein Blick in die Partner-Thesis sicher als lohnend erweisen. Dies gilt insbesondere für den Bereich der Systemarchitektur.

2 Exemplarische Anwendungsfälle

Im Folgenden werden Anwendungsfälle in der Automation beschrieben, an Hand derer die geforderten Funktionen der Middleware erarbeitet werden. Sie dienen auch als Grundlage für die nachfolgenden Konzeptionen. Hierzu gehören die Architektur und für die Applikationsentwickler bereitzustellende Dienste.

2.1 Fertigungslinie

Die industrielle Produktion wird aufgrund von Rationalisierungsmaßnahmen seit geraumer Zeit immer weiter automatisiert. Im Maschinenbau werden hierbei einzelne Fertigungsschritte von Bearbeitungszentren (Zellen) durchgeführt. Zur Abstimmung dient in der Regel ein übergelagertes Produktionsplanungs- und Steuerungssystem (PPS) das unter anderem durch geeignete Verteilung der anstehenden Fertigungsaufträge auf die Zellen eine möglichst hohe Effektivität und Auslastung der Fertigungslinie gewährleisten soll. Darüber hinaus wird zur Überwachung des laufenden Betriebes meist ein Leitstand eingerichtet, der sowohl einen Gesamtüberblick verschaffen soll als auch als Sammelstelle für Betriebsdaten dient:

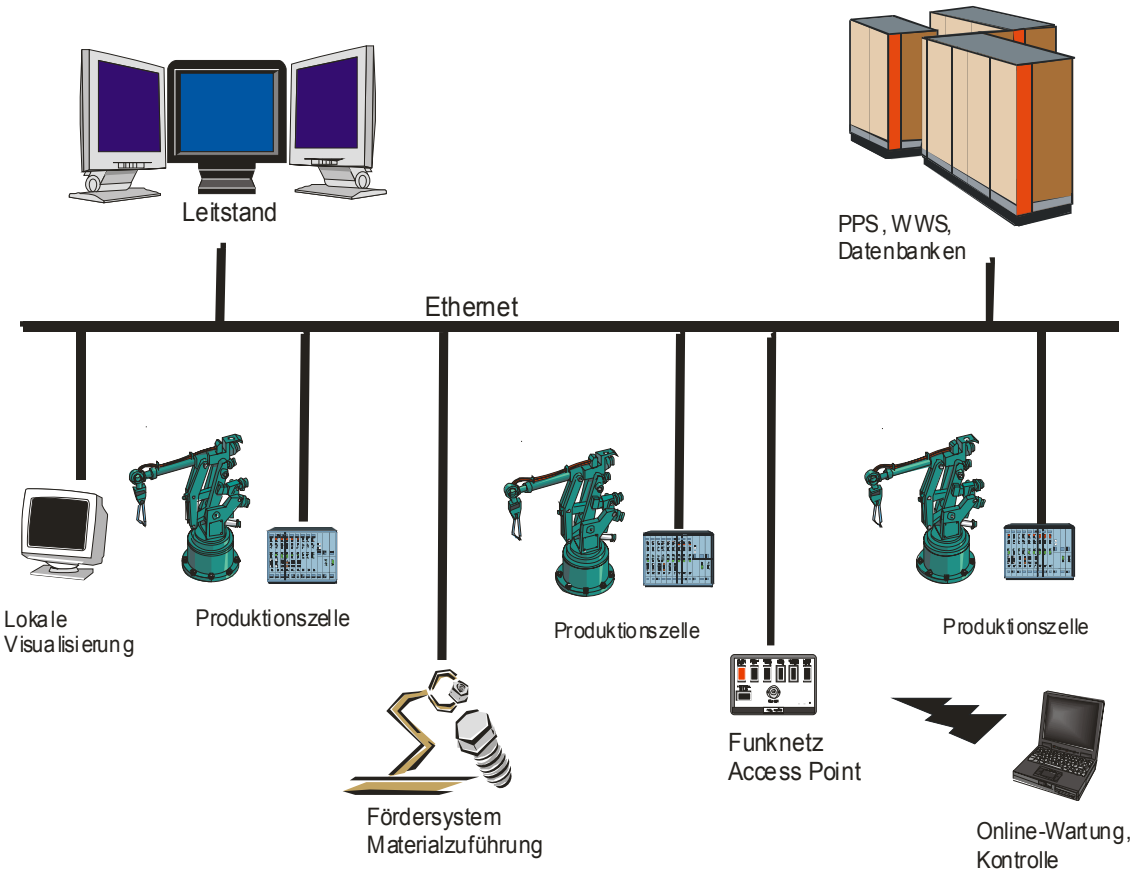

Abbildung 2-1 Produktionslinie

Im Bereich des Sondermaschinenbaus (denn bei den Bearbeitungszentren handelt es sich meist um Spezialanfertigungen) ist in Deutschland eine Vielfalt an Herstellern zu beobachten, die ihre Konkurrenzfähigkeit auch ganz essentiell durch besonderes Spezialwissen in Nischengebieten aufrechterhalten. Als Folge ist eine heterogene Landschaft von Software in Form von Betriebsystemen und Steuerungssystemen anzutreffen. So sind schon auf Feldbus-Ebene eine ganze Anzahl konkurrierender Lösungen vorzufinden. Dies setzt sich ganz besonders auf der Ebene der Steuerungs-Software fort. Hier kommen oft vom jeweiligen Hersteller eigen entwickelte und damit proprietäre Lösungen zum Einsatz. Diese Situation ist an sich als positiv zu bewerten, da zum einen eine heterogene Konkurrenzsituation zu begrüßen ist und zum anderen die in der Automation oft sehr individuellen Anforderungen zum Beispiel an die Funktion von Bearbeitungszentren jeweils speziell zu entwickelnde Software-Lösungen erfordert.

Um jedoch die komplette Produktion automatisieren zu können, müssen die Zellen in der Lage sein miteinander zu interagieren. Dies setzt wiederum eine einheitliche Kommunikation voraus. Ohne erst an die vielfältigen Anforderungen eines PPS oder gar einer kompletten IT-unterstützten Fertigung durch „Computer Integrated Manufacturing“ (CIM) zu denken gibt es hierbei genügend andere wichtige Faktoren der Interoperabilität. Zu nennen sind:

2.1.1 Visualisierung, Überwachung

Die Überwachung einzelner Bearbeitungszentren als auch der Gesamtproduktion erfolgt in der Regel neben der direkten Visualisierung an der Maschine durch einen Leitstand. Die Aufgabe des Leitstands besteht hierbei hauptsächlich in der Schaffung eines Gesamtüberblicks über die aktuell laufende Fertigung. Dies macht eine Aggregation der Informationsflut notwendig. Hierfür wird unter anderem eine Zwischenschicht zur Schaffung einer einheitlichen Schnittstelle für die unterschiedlichen Protokolle der einzelnen Zellen benötigt. Eine weitere Anforderung ist die der Flexibilität: die Einbindung neuer Hardware in Form von Bearbeitungszentren oder Fördersystemen muss sich so einfach wie möglich gestalten.

Die Überwachung der Anlage erfolgt teilweise und ergänzend auch außerhalb eines Leitstandes. Denkbar ist folgendes Szenario: der für die Produktion verantwortliche Mitarbeiter geht durch die Produktionshalle und kann direkt mittels eines mitgeführten Personal Digital Assistant (PDA) oder Notebooks Daten über den Zustand der jeweiligen Maschine anfordern, um kurzfristige Entscheidungen über die Planung von Wartungstätigkeiten oder ähnlichem treffen zu können.

2.1.2 Manufacturing Execution System (MES)

Ziel einer jeden Automation ist die Kostenersparnis. Der Sinn des Einsatzes einer Middleware liegt daher primär in der Unterstützung dieses Zieles, unter anderem durch das Bereitstellen von Auswertungsmöglichkeiten, die unter dem Begriff des MES zusammengefasst werden. Humphrey schreibt dazu:

„Einzelne Maschinen und ganze Produktionslinien müssen mit übergeordneten Geschäftssystemen verbunden werden, um beispielsweise über die Auswertung historischer Daten präzise feststellen zu können, welche Produktionslinie die Endprodukte pünktlich und zu minimalen Kosten liefern kann“. [DHID1104]

Ein MES dient unter anderem als gemeinsame Datenbasis für die Betriebsdatenerfassung (BDE) und die Maschinendatenerfassung (MDE) und hat folgende Aufgaben (Beispiele):

- Laufzeit- und Stückzahlerfassung. Hierbei auch die Erfassung von Stillstandszeiten als Grundlage für Statistiken und Erhöhung der Auslastung.
- Zuordnung von Auftragsdaten sowie Störgründen zu den Betriebsdaten.
- Bedarfsgesteuerte Personaleinsatzplanung inklusive Lohnberechnung. Eine Herausforderung hierbei ist zum Beispiel die Einbeziehung von Springer-Tätigkeiten.
- Im Bereich von CAQ: die Überwachung des Werkzeugverbrauchs.

Für diese Aufgaben werden lückenlose Aufzeichnungen über Auslastung, Fehler- und Ausfallhäufigkeit bei der Bearbeitung der Halbfertigteile sowie die Verfügbarkeit der jeweiligen Zelle an sich benötigt. Folglich muss ein Datenverlust auch bei zeitweiligem Ausfall der Datenbank verhindert werden. Und ganz besonders in diesem Bereich stellt sich die Anforderung nach einer einheitlichen Schnittstelle für die Integration unterschiedlichster Software-Systeme, denn:

“Während sich im Bereich der Organisationssoftware … weitestgehend Standards entwickelt haben, findet man im operativen Bereich der Fertigung eine wilde Ansammlung von Insellösungen für unterschiedlichste Aufgaben.” [ThKoIA42/04]

Zudem liefert ein MES sowohl die Datengrundlage für das PPS-System, als auch für eine Vor- und Nachkalkulation der Produktionskosten:

„Durch die mitlaufende Aufzeichnung der einzelnen Arbeitsgänge kann am Ende festgestellt werden, welche Arbeitszeiten und –kosten auf den entsprechenden Auftrag zu verbuchen sind.“ [UwBoIA42/2004].

2.1.3 Flexibilität

Die Reaktionszeiten auf unvorhergesehene Situationen, wie zum Beispiel der Ausfall einzelner Systeme (Zellen, Steuerungen, Fördertechnik) in der Fertigung, sollten so gering wie möglich gehalten werden. Es wird also eine hohe Flexibilität hinsichtlich von automatischen Umlagerungen der anstehenden Arbeitsaufträge gefordert, zum Beispiel auf geeignete andere Zellen innerhalb der Fertigung. Optimalerweise geschieht dies durch das Gesamtsystem mittels einer Kommunikation zwischen den Zellen und wenn möglich, zunächst ohne Einschalten der übergelagerten Instanz (in der Regel ein PPS-System). So könnte eine Zelle, die in der Produktion parallel geschaltet ist und damit die gleiche Aufgabe erfüllt, eigenständig die offenen Fertigungsaufträge übernehmen. Auch ein Eskalations-Mechanismus ist vorzusehen: ist eine Lösung der Ausnahmesituation auf der Produktionsebene nicht möglich, wird die Situation eine oder mehrere Ebenen höher propagiert. Erfolgt auch dies automatisch, dann ist ein Eingriff durch den Menschen nur noch in seltenen Fällen notwendig, für die keine vorgesehenen Reaktionsweisen definiert sind.

Zur Flexibilität gehört auch der Umgang mit Ausschussteilen in der Fertigung. Für solch fehlerhafte Teile müssen geplante Bearbeitungsschritte in der Fertigungskette storniert werden, was wiederum Auswirkungen auf die Auslastung der betroffenen Zellen hat. Hier ist eine Kommunikation der Zellen untereinander denkbar, die ihre Auslastung optimieren, indem sie freigewordene Kapazitäten melden ohne zunächst eine Rücksprache mit übergelagerten Systemen halten zu müssen.

2.1.4 Programmverwaltung

Eine wichtige Aufgabe eines jeden übergelagerten Software-Systems in einer Produktionslinie ist die zentrale Verwaltung von Programmen, die für die einzelnen Zellen in der jeweils dort eingesetzten Programmiersprache geschrieben wurden. Dies erfordert unter anderem die Bereitstellung einer Versionsverwaltung als auch Suchalgorithmen. Somit bietet sich für diese Aufgaben der Einsatz einer Datenbank an.

2.1.5 Kopplung mit Logistiksystemen

Die Zuführung der Halbfertigteile zu den einzelnen Zellen kann durch ein automatisches Fördersystem ergänzt sein. Hier stellt sich neben der Anforderung an eine Kommunikation mit den Zellen an sich auch die Notwendigkeit einer Zeitsynchronisierung. Zum Beispiel meldet eine Zelle die geplante Fertigstellung eines gerade bearbeiteten Teiles an das Fördersystem und fordert für diesen Zeitpunkt gleichzeitig Nachschub an. Hierbei stellen sich mehrere Anforderungen: zum einen muss diese Meldung persistent sein, darf also bei zwischenzeitlichem Verbindungsabbruch nicht verloren gehen. Andererseits ist ein Ausfall

des Fördersystems nur relevant, wenn es zum geforderten Zeitpunkt nicht wieder betriebsbereit und online ist. Folglich ergibt sich die Notwendigkeit einer Implementierung von ausreichend differenzierten Fehler-Reaktionen sowohl zeitlich als auch in Abhängigkeit von den zu benachrichtigenden Teilnehmern in Netz.

2.1.6 Kopplung mit übergelagerten Verwaltungssystemen

Neben der parallelen Kommunikation zwischen den Zellen und eventuellen Logistik-Systemen ist auch die Kopplung mit der übergelagerten Ebene in Form eines Produktionsplanungs- und Steuerungssystem (PPS) einzubeziehen. So kann aus dem PPS die Anforderung kommen, einen gerade laufenden Produktionsauftrag abzubrechen oder zeitlich umzuplanen. Ein PPS arbeitet hierbei auf einer höheren Abstraktionsebene. So wird die Anforderung einen Produktionsauftrag zu stornieren nur einmal und allgemein der Fertigung gemeldet. Es liegt in der Verantwortung des Fertigungssystems, die jeweils betroffenen Teilsysteme über solch eine Änderung zu informieren. So stellt sich die Anforderung eine Möglichkeit der Definition ineinander verschachtelter Gruppen bereitzustellen, die sowohl rekursiv als auch meldungsabhängig sein sollten. So sendet zum Beispiel das PPS eine Meldung über Änderungen von Produktionsaufträgen an eine bestimmte Zelle. Die Zelle beinhaltet ihrerseits mögliche Subsysteme wie eine Bilderkennung in Form einer intelligenten Kamera, für die die jeweils anstehende Meldung aber nicht unbedingt relevant sein muss.

2.1.7 Anlagenkonfiguration, Backup-Mechanismen

Ein weiteres wichtiges Thema ist die Verwaltung von Anlagendaten wie auch Backup-Mechanismen, die bei Ausfall einer Steuerung deren schnellen Ersatz ermöglichen und hierdurch die Maschinen-Stillstandszeiten minimieren. Das Beispiel einer Fertigungsstrasse für Autos verdeutlicht dies auf besonders anschauliche Weise. Wird die Anlage abrupt zum Stillstand gebracht (zum Beispiel durch Betätigen des Not-Aus-Schalters), verbleiben die jeweiligen Schweiss-Roboter auf ihren gerade eingenommenen Positionen in der Karosserie des jeweiligen in der Produktion befindlichen Autos. Bei dieser Aktion kann die Information über die aktuelle Position des Roboters verloren gehen. Als Folge muss jeder einzelne in dieser Situation befindliche Roboter mittels Handsteuerung von Mitarbeitern einzeln aus der Karosserie navigiert werden. Diese Situation könnte durch den Einsatz einer geeigneten Middleware entschärft werden. Hierfür meldet jeder Roboter während des Fertigungsprozesses ständig seine aktuelle Position an einen Dienst der Middleware. Da sich der Dienst auf einem anderen Rechner im Netz befindet, wird die Ablaufgeschwindigkeit des Roboters an sich nur marginal beeinträchtigt. Voraussetzung ist jedoch eine ausreichend schnelle Verbindung und Reaktionszeit der Middleware an sich.

2.1.8 Extrahierte spezifische Anforderungen

Aufgrund der betrachteten Produktionslinie können folgende spezifische Anforderungen an eine unterstützende Middleware extrahiert werden:

- Die Einführung einer einheitlichen Kommunikation aufgrund der Notwendigkeit ein heterogenes Produktumfeld miteinander zu kombinieren.
- Eine vom Sender unabhängige Verteilung von Informationen zur Erreichung von Unabhängigkeit und Flexibilität.
- Die Verteilung von Teilaufgaben mit dem Ziel einzelne Teilnehmer zu entlasten, als auch eine Aggregation und damit Verallgemeinerung zu erreichen.

2.2 Kommissionieranlage

Ein weiteres Einsatzgebiet ist die Logistik-Automation in den diversen Ausprägungen Kommissionierung sowie unterschiedlichen Arten der Lagerverwaltung.

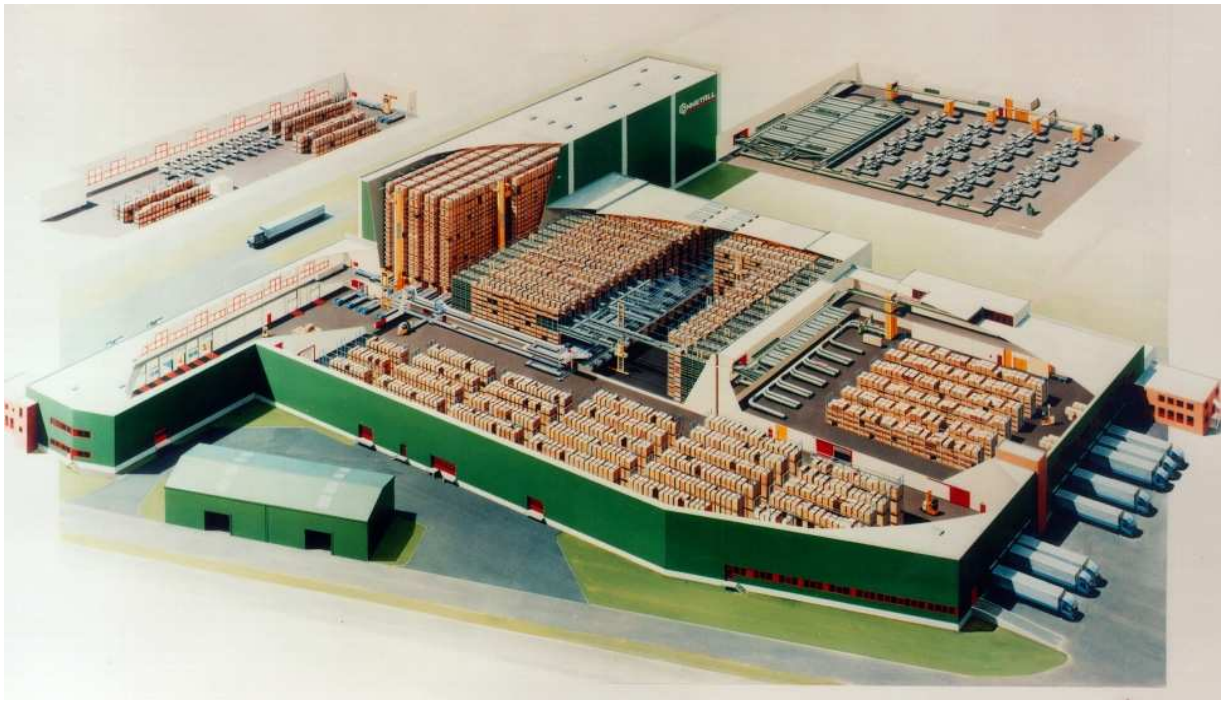

Abbildung 2-2 Automatisiertes Lager [viastore]

So lohnt sich eine komplette Automatisierung nicht in allen Fällen. Während zum Beispiel ein vollautomatisiertes Hochregallager eher für größere Mengen eines eingelagerten Artikels gedacht ist, werden Kleinteile oder auch Artikel in geringen zu pickenden Mengen von Hand kommissioniert. Zu diesem Zweck hat sich der Einsatz von Personal Digital Assistants (PDA's) für die Datenerfassung vor Ort durch den Kommissionierer etabliert, oft auch in Kombination mit einem eingebauten Barcode-Scanner. Das Handheld zeigt ihm dabei den zu bearbeitenden Kommissionierauftrag und führt ihn zu den Lagerorten. Durch Scannen wird die Entnahme (der Pick) bestätigt und per Funk dem Lagerverwaltungssystem mitgeteilt. Es handelt sich also schlussendlich um ein verteiltes System mit auf verschiedenen Ebenen angesiedelten Teilnehmern: PDA, Warenwirtschaftssystem (Host) sowie Leit- und Steuerrechner deren unterschiedliche Hardware und Betriebssysteme durch geeignete Kommunikationstechniken zu kombinieren sind.

2.2.1 Spontaneous Networking

Hierbei ist besonders die Eigenart der dynamischen Verbindungen interessant: Die PDAs werden in der Regel mittels Funk den Kontakt zum Leitrechner suchen. Es kommt nun vor, dass der Kontakt abbricht wenn sich zum Beispiel der Kommissionierer hinter einem besonders hohen Regal befindet, er das Gerät zur Pause ausschaltet oder die Batterie leer ist. Folglich stellt sich die Anforderung, mit solch temporären Verbindung umgehen zu können, um die Persistenz der Datenkommunikation zu gewährleisten.

2.2.2 Lagerorte

Für stärker automatisierte Lager- und Kommissionieranlagen kommen oft Förderstrecken in Form von Rollenförderern zum Einsatz. Ein kritischer Punkt hierbei ist die korrekte Einlagerung der spezifischen Ware. Denn wird der falsche Lagerort verbucht, so gestaltet sich das nachträgliche Auffinden der Ware als enorm aufwendig, wenn nicht sogar unmöglich. Bewirkt darüber hinaus ein essentieller Fehler in der Verbuchung eine generell falsche Einlagerung, kann der Notstand nur durch eine Inventur in Form einer kompletten Auslagerung behoben werden. Solche Verbuchungsfehler treten beispielsweise durch einen nicht geplanten Versatz in der Kommissionierstrecke auf: Die verantwortliche Steuerung merkt sich hierbei am so genannten Identifikationspunkt (I-Punkt) Art und Inhalt des Paketes und schickt dieses auf die Förderstrecke. Die Position des Paketes auf dem Weg zum Lagerort wird mittels Lichtschranken erkannt, wobei die Steuerung sich die Reihenfolge aller auf der Strecke befindlichen Pakete merkt:

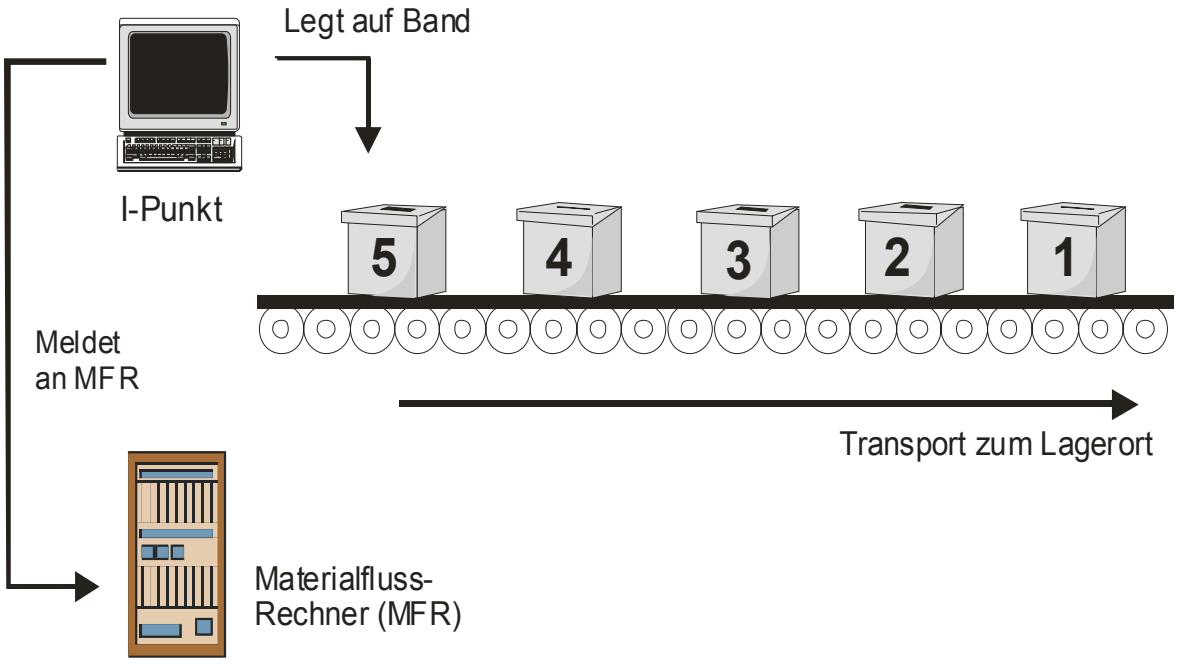

Abbildung 2-3 Kommissionierung: Reihenfolge auf Band

Verliert nun die Steuerung die Kenntnis über die aktuelle Position und Reihenfolge der Pakete (zum Beispiel durch Stromausfall), kann die kritische Situation nur durch Entleeren der Förderstrecke behoben werden. Auch der Einsatz von Identifizierungs-Mechanismen in Form von Transpondern oder RFIDs[1] für die einzelnen Pakete behebt das Problem nur teilweise: sie müssen immer noch zur Identifizierung zumindest an einen weiteren I-Punkt transportiert werden. Eine Middleware könnte für solche Anforderungen den Steuerungen unterstützende Funktionen in Form von Backup-Mechanismen oder nichtflüchtigen Speicher bereitstellen, mittels deren eine Wiederherstellung des letzten Zustandes vor dem Systemausfall möglich ist.

2.2.3 Transaktionen

Der Begriff der Transaktion ist im Bereich der Datenbanken wohl bekannt. Dort dient sie zur Sicherstellung der Datenkonsistenz. So ist die Transaktion einer Banküberweisung erst dann abgeschlossen wenn sowohl das Zielkonto als auch das Quellkonto verbucht worden sind. Schlägt eine dieser Aktionen fehl, dann erfolgt die Herstellung des originalen Zustandes zu Beginn der Transaktion, auch als „Rollback“ bezeichnet. Der Begriff der Transaktion kann jedoch über solche Konsistenzsicherungen hinaus auch auf umfangreichere Abläufe angewendet werden. Durch die beispielhafte Betrachtung der Ein- und Auslagervorgänge in einem Lager für Textilien (Stoffe, Knöpfe, Fäden) soll dies näher verdeutlicht werden:

2.2.3.1 Einlagerung

Ein LKW liefert eine Ladung Stoffballen an. Um die Qualität der angelieferten Ware zu überprüfen werden einige Ballen ausgeschleust um deren Qualität durch Aufspannen auf einer Schau-Wickelmaschine und folgender Sichtprüfung zu verifizieren. Diese Ballen werden als so genannte QS-Ballen (in der Qualitätssicherung befindlich) gekennzeichnet. Da die Vorzone des Lagers für weitere Lieferungen frei gehalten werden muss, wird derweil die Einlagerung der restlichen Ballen initiiert. Nun erweist sich nach einer gewissen Zeit der QS-Ballen als fehlerhaft und es wird die Entscheidung getroffen, die gesamte Lieferung anstatt einzulagern in einen dedizierten Sperrbereich des Lagers zu transportieren. Dies hat folgende Auswirkungen:

- Schon eingelagerte Ballen müssen wieder ausgelagert werden,
- gerade auf der Förderstrecke befindlichen Ballen müssen umgeleitet werden,
- noch in der Vorzone befindliche Ballen müssen eine neue Zieladresse bekommen.

[1] Radio Frequency Identification Devices

Jede dieser Aktionen impliziert eine ganze Anzahl von geänderten und für die Förderstrecke einzuplanenden Transportaufträge, so dass die klassische Umsetzung dieser Anforderung, die Umbuchung von einer zentralen Instanz aus, sich als recht aufwendig erweist. Nun kann der Vorgang „Einlagerung aller Stoffballen der Lieferung“ auch als eine Transaktion angesehen werden, die durch die Entscheidung für die Umlagerung ein „Rollback mit neuem Ziel“ erfährt und ihrerseits selbständig alle notwendigen Aktionen durchführt.

2.2.3.2 Auslagerung

Transaktionen können auch im Bereich der Kommissionierung auftreten, wie folgendes Beispiel erläutern soll. Ein Produktionsauftrag benötigt eine durch die Stückliste und die Anzahl der herzustellenden Fertigteile festgelegte Anzahl an Rohwaren. Diese Rohwaren müssen zusammengestellt und als Ganzes ausgelagert werden. So werden zum Beispiel für eine Anzahl zu produzierender Hosen eine definierte Menge Stoff, Faden, Knöpfe, Einlagen und anderes benötigt. Das übergelagerte Warenwirtschaftssystem sendet daher einen Auslagerauftrag mit den geforderten Mengen an das Lager, der an sich als eine Transaktion betrachtet werden kann. Das Lagerverwaltungssystem erzeugt wiederum die benötigten Transportaufträge, zum Beispiel für die Auslagerung des Stoffes aus dem Hochregallager sowie die zugehörigen Knöpfe aus dem Kleinteilelager. Während der nebenläufig ablaufenden Auslagervorgänge trifft nun eine nicht vorhergesehene Situation ein: Knöpfe fehlen, der Stoff wird beschädigt oder ähnliches. Da der Bedarf für den Produktionsauftrag nur komplett ausgeliefert werden soll, muss der gesamte Auslagervorgang gestoppt und eventuell schon ausgelagerte Waren wieder eingelagert werden. Die Auslagerung wird also storniert was einem klassischen Rollback der Transaktion entspricht und, wie schon beim Einlagerbeispiel, die Generierung eines Satzes geeigneter Transportaufträge für die Wiedereinlagerung erfordert.

Besonders das letzte Beispiel verdeutlicht die unterschiedlichen Bedeutungen eines „Rollbacks“: Das Ziel eines klassischen Rollbacks, wie es bei Datenbanken definiert wird, ist es, den Urzustand genau wieder herzustellen. Nicht so die Rückführung der schon ausgelagerten Waren in das Lager. Hier ist es nicht das Ziel, die Waren an genau die gleiche Stelle wieder einzulagern, von der sie entnommen worden sind. Dieser Lagerort ist womöglich zwischenzeitlich durch eine unabhängige Einlagerung schon wieder besetzt. Die Definition „Ausgangszustand“ ist somit auf einer höheren Ebene angesiedelt: die Ware ist unabhängig vom tatsächlichen Lagerort wieder in den Zustand „am Lager“ zu bringen und damit für weitere Entnahmen verfügbar zum machen.

Bietet nun die Middleware generelle Funktionen an, mit deren Hilfe auf abstrakter Ebene Entscheidungen und Reaktionen definiert werden können, kann die softwaretechnische Implementierung solcher Abläufe erheblich vereinfacht werden.

2.2.4 Optimierungen

Prinzipiell begrenzt die jeweils eingesetzte Fördertechnik die maximale Geschwindigkeit des Systems. So ist die Zeitspanne für ein Doppelspiel in einem Hochregallager - zum Lagerort hinfahren, Ware holen und zum Auslagerpunkt bringen - hauptsächlich von den mechanischen Randbedingungen des Regalbediengerätes (RBG) wie Beschleunigung und maximale Geschwindigkeit abhängig. Um trotzdem einen möglichst hohen Durchsatz zu erreichen, gibt es verschiedene Ansätze zur Optimierung. So kann durch geeignete Kombination der Fahraufträge innerhalb eines Doppelspiels eine kombinierte Ein- und Auslagerung erfolgen. Auch ist sicherzustellen, dass die Zuführung zum RBG ohne Zeitverlust erfolgt.

2.2.5 Anlagenauslastung

Die Kapazität eines Logistik-Systems ist grundsätzlich begrenzt. Bei Überlastung wird ein Stau erzeugt, den es so weit wie möglich zu vermeiden gilt. Eine Möglichkeit ist, so genannte Puffer-Bahnhöfe einzurichten, in denen Ware bei Überlastung zwischengelagert wird. Eine Überlast in gesonderten Bereichen kann jedoch auch aufgrund von Meldungen vorgelagerter Transportstrecken vorherberechnet werden, so dass geeignete Maßnahmen getroffen werden können bevor die Anlage aufgrund eines Staus stehen bleibt. Dies kann beispielsweise durch eine direkte Kommunikation der einzelnen Steuerungen untereinander realisiert werden, die - ähnlich einem Agentensystem - Situationen melden und über alternative Routen verhandeln.

Solche Aufgaben stellen einen Teil des so genannten Materialflussrechners (MFR) dar. Hierbei handelt es sich um ein Programm, das unter anderem durch die Planung und Verteilung der Fahraufträge auf die Fördertechnik eine möglichst optimale Auslastung der Anlage sicherstellen soll. Zur Unterstützung dieses Programms kann die Middleware zusätzlich Dienste bereitstellen.

2.2.6 Echtzeitanforderungen (Realtime)

Bei bestimmten Abläufen in der Automation können spezielle Anforderungen an ein Echtzeit-Verhalten gestellt werden. Unter Echtzeit ist hierbei die Rechtzeitigkeit, also die zugesicherte Reaktion innerhalb eines vorgegebenen Zeitraums zu verstehen. Sie ist folglich nicht

unbedingt mit dem Begriff „schnelle Reaktion“ gleichzusetzen. Man unterscheidet zwischen harten und weichen Echtzeitanforderungen. So schreibt Burns:

„Hard real-time systems are those where it is absolutely imperative that responses occur within the specific deadline. Soft real-time systems are those where response times are important but the system will still function correctly if deadlines are occasionally missed.” [BWRS93]

Da die Anforderungen an ein hartes Echtzeitsystem sehr hoch sind, müssen sämtliche Ebenen echtzeitfähig gestaltet werden. Dies schließt neben einem geeigneten Transportprotokoll für die Übertragungsschicht ein echtzeitfähiges Betriebssystem bis hin zu echtzeitfähigen Applikationen ein. Folglich reicht es nicht aus wenn die Middleware als solches als „echtzeitfähig“ tituliert wird. Weiche, und damit unkritischere Echtzeitanforderungen hingegen können zumindest zum Teil auf konventionellen Netzen und Betriebssystemen umgesetzt werden. Zur Verdeutlichung wird dazu am folgenden Beispiel der rechtzeitigen Ausschleusung eines Paketes exemplarisch eine weiche Echtzeitanforderung veranschaulicht[2]. Für die Versendung von Büchern wird eine automatische Kommissionieranlage eingesetzt. Diese besteht aus einem inneren Kreis für die mit den Büchern zu füllenden Kartons und Stationen, an denen jeweils eine Teilmenge aller zu versendenden Bücher positioniert sind:

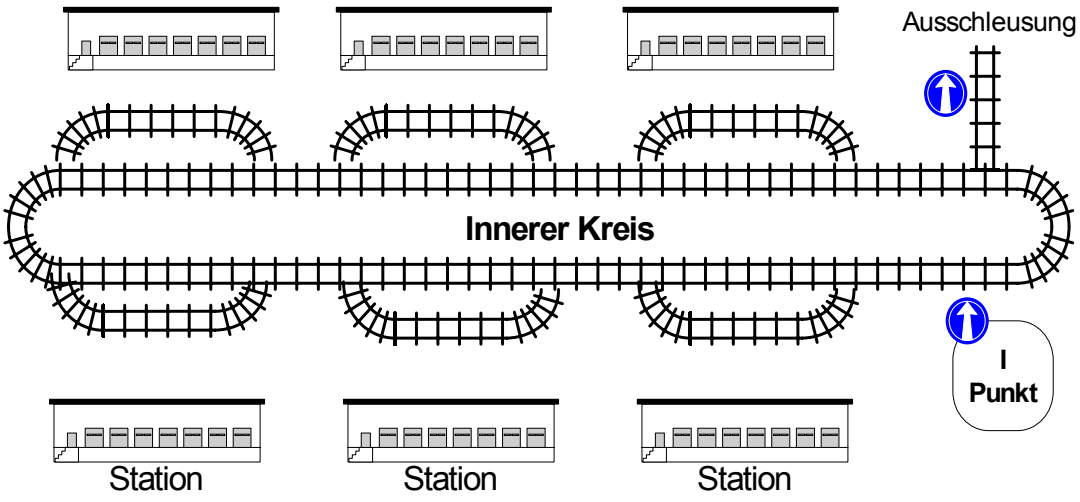

Abbildung 2-4 Kommissionieranlage Buchversand

[2] Beim vorliegenden Beispiel handelt es sich um einen realen Vorfall innerhalb eines Projekts bei dem der Autor als Software-Entwickler beteiligt war.

Am I-Punkt wird ein leerer Karton auf das Tablar gesetzt und ein Kommissionierauftrag sowohl mit dem Karton als auch dem Tablar verknüpft[3]. Das Tablar geht auf die Reise im inneren Ring und soll überall dort ausgeschleust werden, wo dem Auftrag entsprechend Bücher zu picken sind. Ist der Auftrag komplett befriedigt, wird der volle Karton aus dem inneren Kreis ausgeschleust und der Versandabteilung zugeführt. Hierfür ist kurz vor der Station ein Detektor installiert, der die Identifikation des Tablars von seinem Transponder ausliest:

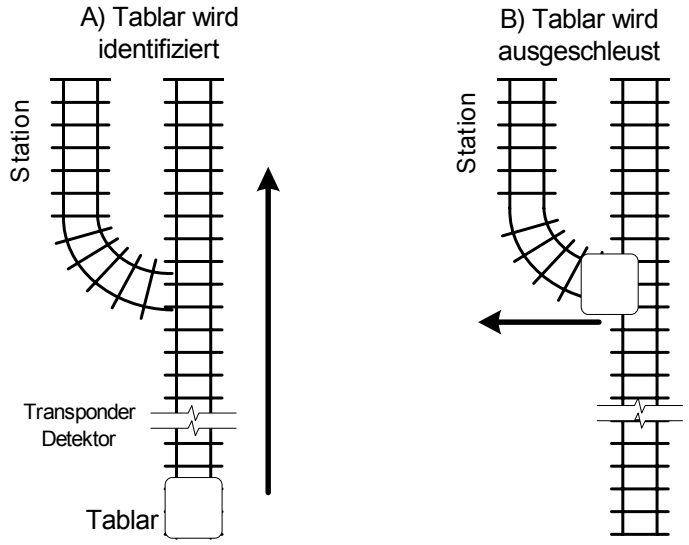

Abbildung 2-5 Ausschleusung Tablar

Jedes Tablar identifiziert sich etwa einen Meter vor der Ausschleusungsstelle mittels Auslesen eines Transponders. Die Steuerung startet darauf hin eine Anfrage an den Leitrechner ob das Tablar an die aktuelle Station ausgeschleust werden soll. Der Leitrechner überprüft den Status des Auftrages, vergleicht die noch zu pickenden Bücher mit der an der anfragenden Station befindlichen Teilmenge und beantwortet daraufhin die Anfrage. Die hier beschriebene Kommunikation muss abgeschlossen sein bevor das sich mit einer definierten Geschwindigkeit bewegende Tablar die Ausschleusungsstelle passiert. Die Middleware muss daher auch in Zeiten starker Netzauslastung für diesen Ablauf eine gesicherte Antwortzeit gewährleisten. Wird das Zeitfenster verpasst, sind die Auswirkungen zwar unerwünscht aber unkritisch im Hinblick auf das Funktionieren der Anlage an sich: das Tablar bekommt nach einer Umkreisung wieder die Möglichkeit der Ausschleusung. Somit handelt es sich um eine weiche Echtzeitanforderung.

[3] In der Logistik wird dieser Vorgang auch „verheiraten“ genannt.

Ausschnitt der Kommissionieranlage:

Das Tablar im Vordergrund wird gerade an die Station ausgeschleust und folgt dem schon in der Station befindlichen Tablar.

Im inneren Kreis sind weitere Tablare zu sehen, die eventuell die Weiche verpasst haben.

(Da das Foto während der Inbetriebnahme aufgenommen wurde, fehlen die zu bepackenden Kartons auf den Tablaren).

Abbildung 2-6 Bild Tablar-Weiche

2.2.7 Extrahierte spezifische Anforderungen

Aufgrund der betrachteten Logistiksysteme können zusätzlich zu Kapitel 2.1 folgende spezifische Anforderungen für diesen Bereich an eine unterstützende Middleware extrahiert werden.

- Die Unterstützung von dynamischen Verbindungen einzelner Kommunikationspartner.
- Funktionen zur Unterstützung von Persistenz damit systemrelevante Daten auch bei einem Teilausfall einzelner Teilnehmer nicht verloren gehen.
- Die Bereitstellung von Funktionen zur Kapselung von Einzelaufgaben zu einer Transaktion.
- Unterstützung von Optimierungsmechanismen in Form von Analyse-Werkzeugen, um z.B. Auslastungen von Teilbereichen der Anlage zu ermitteln.
- Unterstützung weicher Echtzeitanforderungen indem die Möglichkeit von Priorisierungen der Kommunikation bereitgestellt wird.