

Forschungsbericht

**KoDeMat - Befähigung von KMU zur kollaborativen
Planung und Entwicklung heterogener, dezentral
gesteuerter Materialflusssysteme**

ten Hompel, M. · Günthner, W. A. · Roidl, M. · Kipouridis, O.

Forschungsbericht

Zu dem IGF-Vorhaben

KoDeMat - Befähigung von KMU zur kollaborativen Planung und Entwicklung heterogener, dezentral gesteuerter Materialflusssysteme

der Forschungsstellen

TU Dortmund, Lehrstuhl für Förder- und Lagerwesen

TU München, Lehrstuhl für Fördertechnik Materialfluss Logistik

Das IGF-Vorhaben **17391N** der Forschungsvereinigung Bundesvereinigung Logistik (BVL) e. V. würde über die



im Rahmen des Programms zur Förderung der Industriellen Gemeinschaftsforschung (IGF) vom Bundesministerium für Wirtschaft und Technologie aufgrund eines Beschlusses des Deutschen Bundestages gefördert.

Gefördert durch:



Bundesministerium
für Wirtschaft
und Technologie

aufgrund eines Beschlusses
des Deutschen Bundestages

Herausgegeben von:
Prof. Dr. Michael ten Hompel
Lehrstuhl für Förder- und Lagerwesen FLW
Joseph-von-Fraunhofer-Str. 2-4
44227 Dortmund
<http://www.flw.mb.tu-dortmund.de/>

Prof. Dr.-Ing. Dipl.-Wi.-Ing. Willibald A. Günthner
fml – Lehrstuhl für Fördertechnik Materialfluss Logistik
Technische Universität München
Boltzmannstr. 15
85748 Garching
www.fml.mw.tum.de

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, der Entnahme von Abbildungen, der Wiedergabe auf photomechanischem oder ähnlichem Wege und der Speicherung in Datenverarbeitungsanlagen bleiben – auch bei nur auszugsweiser Verwendung – vorbehalten.

Copyright © FLW – Lehrstuhl für Förder- und Lagerwesen
Copyright © fml – Lehrstuhl für Fördertechnik Materialfluss Logistik

Printed in Germany 2015

Zusammenfassung

Im Forschungsprojekt *KoDeMat - Befähigung von KMU zur kollaborativen Planung und Entwicklung heterogener, dezentral gesteuerter Materialflusssysteme* werden die Grundlagen für einen kollaborativen Engineeringprozess im Bereich der Neu- und Umplanung fördertech-nischer Anlagen mit mehreren beteiligten Unternehmen geschaffen.

Das Forschungsprojekt lässt sich in den breiteren Kontext der fortschreitenden Automatisierung der Logistik- und Produktionssysteme einordnen. Mit seiner praktischen Ausrichtung verfolgt es die Zielsetzung der Einführung von Methoden, wie sie in der Entwicklung hin zu einer *Industrie 4.0* erforderlich werden. Die im Forschungsprojekt erarbeiteten Ergebnisse ermöglichen es KMU ihre existierenden Systeme weiterzuentwickeln und an die Rahmenbedingungen einer zukünftigen *Industrie 4.0* anzupassen.

Im Forschungsprojekt wurde ein kollaborativer Engineeringprozess gestaltet, der eine unternehmensübergreifende Planung, Realisierung und Inbetriebnahme von komplexen, dezentral gesteuerten Intralogistiksystemen sowie deren Betrieb effizient ermöglicht. Zentrale Bestandteile dieses Prozesses sind die Lösung der Schnittstellenproblematik bei der Zusammenschaltung mehrerer unabhängiger intralogistischer Subsysteme sowie die Schaffung einer digitalen Abbildung des Gesamtsystems über den Prozessverlauf.

Zur Prozessunterstützung wurde eine leistungsfähige, offene Kollaborationsplattform entwickelt und implementiert. Über diese Plattform wird eine einheitliche Datensynchronisierung sichergestellt, die eine Planung, Visualisierung und virtuelle Inbetriebnahme sämtlicher am Materialflusssystem beteiligter Anlagenteile ermöglicht.

Der Nachweis der Funktionsfähigkeit und Anwendbarkeit von Kollaborationsprozess und -plattform wurde über ein Demonstrationsprojekt geführt, in welchem die Zusammenschaltung und verteilte Inbetriebnahme der realen Versuchsanlagen als virtuelles Gesamtsystem der beiden beteiligten Forschungsstellen durchgeführt wurde. Darüber hinaus wurde die Erweiterbarkeit und Flexibilität der Plattform durch die Integration weiterer logistischer Systeme gezeigt. Die Integration hochmoderner VR-Brillen und Multitouch-Tische für die 3D-Visualisierung ermöglicht vollkommen neue Planungsprozesse mit

vielen beteiligten Benutzern. Komplexe Intralogistiksysteme können somit gemeinsam von mehreren Herstellern/Lieferanten antizipiert, getestet und virtuell in Betrieb genommen werden.

Auf diese Weise verkürzt sich die Planungsphase durch effizientere Abstimmungsprozesse zwischen den beteiligten Unternehmen. Auch lassen sich Verzögerungen, die während der Inbetriebnahme durch Schnittstellenprobleme zwischen einzelnen Anlagenteilen auftreten können, vermeiden. Zeiteinsparungen in diesen Projektphasen führen zu Kostenersparnissen für den Anlagenbetreiber. Für die Hersteller/Planer ergibt sich die Möglichkeit, das eigene Geschäftsfeld um Projekte zu erweitern, die bisher aufgrund ihrer Größe/Komplexität nicht durchführbar waren.

Inhaltsverzeichnis

1	Forschungsthema	1
2	Wissenschaftlich-technische und wirtschaftliche Problemstellung	7
2.1	Digitale Fabrik	7
2.2	Industrie 4.0	13
2.3	Collaborative Engineering	14
2.4	Verteilte und kollaborative Softwareentwicklung	18
2.5	Planung und Umsetzung technischer Logistiksysteme	21
3	Forschungsziel und angestrebte Ergebnisse	31
3.1	Forschungsziel	31
3.1.1	Angestrebte Forschungsergebnisse	31
3.1.2	Innovativer Beitrag der angestrebten Forschungsergebnisse	33
3.2	Lösungsweg zur Erreichung des Forschungsziels	34
4	Design eines kollaborativen Engineeringprozesses zur Planung heterogener Materialflusssysteme	39
4.1	Systemarchitektur eines agentengesteuerten Materialflusssystemes	43
4.2	Fazit	45
5	Anforderungen an die Kollaborationsplattform	47
5.1	Anwendungsfall	48
5.1.1	Akteure	48
5.1.2	Referenzszenario	49
5.2	Geschäfts- und Benutzeranforderungen	51
5.3	Funktionale Anforderungen an die Visualisierung	52
5.4	Funktionale Anforderungen an die Schnittstellen- und Versionsverwaltung	54
5.5	Nicht-funktionale Anforderungen	55
6	Spezifikation und Implementierung einer kollaborativen Softwareplattform	57
6.1	Softwarearchitektur	57

6.2	Verteilung der Daten	59
6.2.1	Synchronisierung der Onlinedaten	60
6.2.2	Synchronisierung und Versionsverwaltung der Off- linedaten	65
6.3	Ermöglichung einer einheitlichen Visualisierung von 3D- Objekten	70
6.4	Importierung / Exportierung von Daten	71
6.5	Fazit	73
7	Implementierung eines Planungswerkzeugs mit 3D- Visuali- sierung	75
7.1	Kollaborative Planung und Erstellung eines Anlagelay- outs	75
7.2	Planung virtueller Integrationstests	78
7.3	Visualisierung realer Materialflusssystemen	84
7.4	Technische Herausforderungen bei der synchronen kol- laborativen Bearbeitung	89
8	Implementierung der Schnittstellenverwaltung	93
8.1	Agentenorientierte Konfiguration von Schnittstellen zwi- schen Subsystemen	94
8.2	Schnittstellenverwaltung als temporäre Agentenkomm- unikationsplattform	95
8.3	Kollaborative Schnittstellendefinition	96
8.4	Automatische Generierung von Agentenkonfigurations- dateien	98
8.5	Fazit	98
9	Kopplung existierender Demonstratoren	101
9.1	Modellierung eines virtuellen Gesamtsystems	104
9.2	Anbindung der agentenbasierten Steuerungen	107
9.3	Anbindung der Simulationsumgebungen	111
9.4	Definition der Testläufe	111
9.5	Kollaborative Definition der Auftragstelegramme und der Lastübergabeschnittstelle	114
9.6	Verteilter Inbetriebnahmetest des Gesamtsystems	116
9.7	Fazit	117

1 Forschungsthema

Im Forschungsvorhaben *KoDeMat - Befähigung von KMU zur kollaborativen Planung und Entwicklung heterogener, dezentral gesteuerter Materialflusssysteme* werden die Grundlagen für einen kollaborativen Engineeringprozess im Bereich der Neu- und Umplanung fördertechischer Anlagen mit mehreren beteiligten Unternehmen geschaffen. Diesen Prozess unterstützt eine im Projektverlauf konzeptionierte und implementierte offene Kollaborationsplattform. Diese Plattform sorgt für eine einheitliche Datenbasis und ermöglicht die Emulation sämtlicher am Materialflusssystem beteiligter Anlagenteile. Der Fokus liegt dabei auf einer ganzheitlichen Materialflussteuerung.

Die Entwicklung dezentraler Steuerungskonzepte hat sich in den letzten Jahren als ein Schwerpunkt der angewandten Forschung im Bereich automatisierter Materialflusssysteme etabliert (vgl. [HW07], [FM09], [OVFK10]). Mittlerweile existieren erste industrielle Prototypen und Produkte dezentral gesteuerter Intralogistiksysteme (z. B. Dematic Multishuttle, Lanfer THINGtelligence). Die von diesem neuen Systemansatz erwarteten Vorteile liegen in einer verbesserten Skalierbarkeit und Anpassungsfähigkeit sowie in der angestrebten Wiederverwendbarkeit von Systemkomponenten. Wirtschaftliche Vorteile sollen in diesem Zusammenhang durch verkürzte Inbetriebnahmezeiten und eine aufwandsarme Erweiterbarkeit der Materialflussteuerung erzielt werden.

Das *Internet der Dinge* (IdD) bezeichnet eines dieser neuartigen, dezentralen Steuerungsparadigmen für innerbetriebliche Materialflusssysteme (vgl. [GHM10]). Auf Basis von RFID-Technologie und Softwareagenten übernimmt das zu transportierende Gut selbst die steuernde Rolle und nutzt die Transportdienste sowie weitere Dienste einer Förderanlage. Auftragsrelevante Informationen befinden sich prozessnah auf dem RFID-Tag oder Embedded-System des entsprechenden Transporthilfsmittels und geben den abzuarbeitenden Workflow vor. Das Konzept sieht zudem eine Verteilung von Steuerungshard- und software auf sogenannte Fördertechnikmodule vor. Diese Verteilung unterstützt eine hohe Modularität der Steuerung.

Ein weiteres grundlegendes Konzept ist die *Digitale Fabrik*, da hier die Rahmenbedingungen festgelegt werden, in welchem zukünftige lo-

gistische Systeme geplant und betrieben werden (vgl. Abschnitt 2.1). In dem aktuellen Prozess der Wandlung hin zu einer *Industrie 4.0* wird die *Digitale Fabrik* in die reale Fabrik integriert und es entsteht eine *Smart Factory* (vgl. 2.2). Hier kommunizieren Produkte, Maschinen und Ressourcen direkt miteinander. Es entsteht ein Umfeld, das aus intelligenten, sich selbst steuernden Objekten besteht. In diesen Cyber-Physischen Systeme (CPS) wird die Vernetzung der dezentralen intelligenten Objekte durch die flächendeckende und wirtschaftliche Verfügbarkeit der technischen Infrastruktur in Form von industriell einsetzbaren Funk- und Internetverbindungen ermöglicht. Nach dezentralen Steuerungskonzepten geplante Intralogistiksysteme werden zukünftig in diesen Kontext der *Industrie 4.0* als Kernbestandteil eingebettet sein und für den Materialfluss der intelligenten Objekte zuständig sein. Zukünftige Projekte werden durch die Vernetzung vieler Subsysteme mehr Schnittstellen aufweisen und so erheblich komplexer werden.

Die Einführung der dezentralen Steuerungskonzepte wirkt sich auch auf den Entwicklungsprozess fördertechnischer Anlagen aus: Die Aufwände für Planung, Realisierung und Test reduzieren sich im Vergleich zu heutigen Materialflusssystemen, dafür fällt bei der projektunabhängigen Entwicklung von Hardware- und Softwaremodulen ein erhöhter Aufwand an. Die Gründe für diese Verschiebung liegen in der hohen Wiederverwendbarkeit der modularen Mechanik und Steuerungslogik sowie im Einsatz moderner und flexibler Kommunikationstechnologien und Protokolle, welche Parametrierung, Inbetriebnahme und Test solcher Systeme erleichtern. Auch beim Umbau oder bei der Erweiterung einer Logistikanlage lässt sich der Aufwand dank des modularen Aufbaus verringern. Dezentral gesteuerte Materialflusssysteme bieten somit neben zahlreichen funktionalen Vorteilen - wie einer erhöhten Systemflexibilität und Robustheit - auch ein großes wirtschaftliches Potenzial durch Einsparungen in der Planungs- und Inbetriebnahmephase von Materialflusssystemen (vgl. Abbildung 1.1, [GCK08]).

Gerade für heterogene Großprojekte (vgl. Abbildung 1.2), in welchen durch mehrere Hersteller neue Fördertechniksysteme in ein beim Kunden existierendes Materialflusssystem integriert werden, bedarf es geeigneter kollaborativer Engineeringkonzepte um diese Potenziale zu heben. Einerseits besteht die Gefahr, dass durch eine unzureichende Transparenz die Zusammenarbeit der beteiligten Unternehmen be-



Abbildung 1.1: Vergleich der projektspezifischen Aufwände für Planung, Realisierung und Test bei Materialflussanlagen mit zentraler Steuerung (l.) und dezentral gesteuerten Anlagen (r.) ([Kuz10], S.9)

hindert wird. Andererseits kann der Grad der Wandlungsfähigkeit der eingesetzten Subsysteme bei der Integration in das Gesamtsystem verloren gehen.

Die aktuelle Bearbeitung gemeinschaftlicher Intralogistikprojekte findet in einer Vielzahl von Medien statt und ist zuletzt geprägt von einer großen Anzahl an E-Mails, die zwischen verschiedenen Projektbearbeitern verschickt werden. Dies führt zu Medienbrüchen: Die Informationen zur Entwicklung der Materialflusssteuerung finden sich in Textverarbeitungsdateien, Programmdateien, speziellen Simulationsprojekten, usw. Der daraus folgenden Intransparenz kann nur mit einem erhöhten organisatorischen Aufwand begegnet werden, der zu den hohen Kosten der Softwareentwicklung beiträgt. Dieser Effekt verstärkt sich erfahrungsgemäß mit einer höheren Anzahl an Projektbeteiligten, einem größeren räumlichen Abstand der Projektgruppen und einem zeitlichen Versatz in der Durchführung der Arbeitsschritte.

Zudem wird die Transparenz innerhalb der gesamten Produktionskette durch die Verfügbarkeit von in Echtzeit anfallenden Daten steigen.

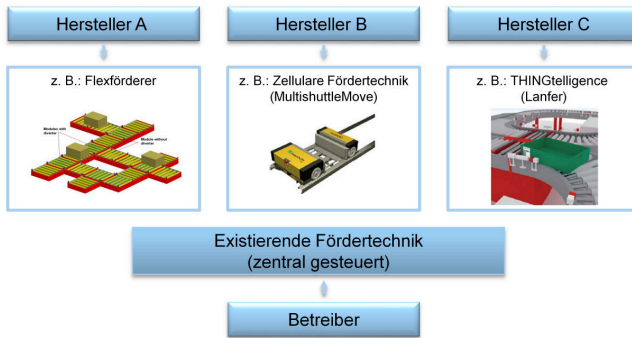


Abbildung 1.2: Beispielhaftes heterogenes Großprojekt

Dies erfordert ein effektives und durchgängiges Collaborative Engineering über den gesamten Lebenszyklus eines Produktes und damit über die horizontalen und vertikalen Verknüpfungen der eingebetteten Produktionssysteme. Bestehende Softwarelösungen für ein Collaborative Engineering eignen sich nicht für den hier adressierten Anwendungsfall.

Im vorliegenden Forschungsprojekt werden die genannten Problemfelder adressiert. Der Schwerpunkt liegt auf der Nutzung und Erweiterung etablierter kollaborativer Techniken aus dem Bereich der Softwareentwicklung. Dabei werden Erfahrungen und Potenziale aus dem BMBF-Projekt *Internet der Dinge* (vgl. [GHM10]) auf den Planungs- und Inbetriebnahmeprozess intralogistischer Anlagen übertragen. Der grundlegende Ansatz ist es, eine Software zu erstellen, deren Softwarebibliotheken auf frei verfügbaren Standardkomponenten basieren. Diese Softwareentwicklung wird als Open-Source-Projekt zur Verfügung gestellt. Eine modulare Softwareentwicklung ermöglicht in diesem Zusammenhang, dass die entwickelten Komponenten auch einzeln anwendbar sind und sich somit einfach, auch in Teilen, in ein existierendes Softwaresystem integrieren lassen.

Dieser Ansatz optimiert mehrere Aspekte der Entwicklung von Materialflusssteuerungen zugleich. Zum einen wird die Wirtschaftlichkeit durch die niedrigeren Kosten bei Abstimmung und Kollaboration gesteigert, zum anderen sorgt eine automatische Schnittstellenverwal-

tung in Kombination mit einer 3D-Visualisierung und Emulation für eine Steigerung der Prozessgeschwindigkeit und Qualität der Software. Die Lösung dieser umfangreichen interdisziplinären Problemstellung erfolgt im Rahmen dieses kooperativen Forschungsprojekts unter Beteiligung der Lehrstühle für Förder- und Lagerwesen (FLW, Lehrstuhlinhaber Prof. Dr. Michael ten Hompel) der Technischen Universität Dortmund und für Fördertechnik Materialfluss Logistik (fml, Lehrstuhlinhaber Prof. Dr. Willibald A. Günthner) der Technischen Universität München. Beide Lehrstühle verfügen über langjährige Erfahrung in der Konzeption, Planung und Realisierung unternehmensübergreifender Materialflusssysteme und deren feldnaher Steuerung (vgl. [GR00], [HSL04], [GW04], [Hom06], [GHW06]).

Dem AiF-konformen Ansatz folgend werden die Problem- und Fragestellungen aus der Praxis abgeleitet. Es findet keine unternehmensspezifische Gewichtung der Aufgabenstellung statt. Dies garantiert eine Gleichbehandlung aller Interessengruppen. Erarbeitete Lösungen stehen allen interessierten Unternehmen zur Verfügung. Anhand des Demonstrators wird lediglich ein Einsatzfall untersucht, der die Grundlage für weitere Entwicklungen bildet.

2 Wissenschaftlich-technische und wirtschaftliche Problemstellung

In den nachfolgenden Abschnitten werden der Stand der Forschung in den Bereichen der *Digitalen Fabrik* (Abschnitt 2.1), des *Kollaborativen Engineerings* (Abschnitt 2.3), der *Planung und Umsetzung technischer Logistiksysteme* (Abschnitt 2.5) und der *Verteilten und kollaborativen Softwareentwicklung* (Abschnitt 2.4) vorgestellt und Anknüpfungspunkte für die Problemlösung im vorliegenden Forschungsprojekt aufgezeigt.

2.1 Digitale Fabrik

Das vorliegende Forschungsprojekt lässt sich in den breiteren Kontext der fortschreitenden Automatisierung, nicht nur im Bereich der Logistik, sondern auch der Produktionssysteme einordnen. Mit seiner praktischen Zielsetzung verfolgt es die Einführung von Methoden, wie sie in der Entwicklung hin zu einer *Digitalen Fabrik* verwendet werden. Daher soll im folgenden Abschnitt nicht nur der Stand der Technik, sondern auch die historische Entwicklung der *Digitalen Fabrik* betrachtet werden. Dieser Ansatz veranschaulicht die Zielsetzung des KoDeMat-Forschungsprojekts als logische Fortführung dieser breiteren Forschungsrichtung. Die im Forschungsprojekt erarbeiteten Ergebnisse ermöglichen es KMU (Kleine und mittlere Unternehmen (KMU)) ihre existierenden Systeme weiterzuentwickeln und an die Rahmenbedingungen einer zukünftigen *Digitalen Fabrik* anzupassen.

Der Begriff *Digitale Fabrik* wird erstmalig 1998 in Zusammenhang mit seiner heutigen Bedeutung verwendet (vgl. [Spi12]). Das große Interesse am Konzept der Digitalen Fabrik innerhalb von Politik, Forschung und Wirtschaft führt zu einer Vielfalt von Definitionen und Interpretationen in der Literatur. Im Folgenden werden die gängigsten Definitionen in chronologischer Reihenfolge vorgestellt, da sich im direkten Vergleich und in der zeitlichen Entwicklung weitere Unterscheidungsmerkmale ergeben.

Dem Konzept der Digitalen Fabrik wird bei der unternehmensübergreifenden Verknüpfung von Simulationstechnologien in der Produktionsplanung eine wesentliche Bedeutung zugewiesen. Dafür ist es je-

doch erforderlich, dass alle Vorgänge und Bereiche miteinander verknüpft werden, die an der Simulation von Komponenten, Fertigungszellen, Abläufen und Geschäftsprozessen beteiligt sind. Vorteile der Digitalen Fabrik entstehen in Form von Synergieeffekten durch die abteilungsübergreifende Datennutzung (vgl. [Spi12]).

Die Digitale Fabrik wird auch als eine rechnerunterstützte Abbildung aller angebotenen Leistungen sowie der zur Herstellung notwendigen Ressourcen und Prozesse einer Fabrik definiert. Das erzeugte virtuelle Modell kann sowohl mit Planungsdaten, als auch mit realen Daten betrieben werden. Demnach liegt das Einsatzgebiet in der Absicherung in der Planung und im laufenden Betrieb eines Unternehmens. Die Bestandteile der Digitalen Fabrik bestehen aus Querschnittsfunktionen, Geschäftsprozessen, prozessrelevantem Wissen sowie aus baulichen, technischen und organisatorischen Strukturen. Aufgrund der umfangreichen Bestandteile liegt eine große Problemstellung in der virtuellen Modellbildung (vgl. [Wes01]).

Das Hauptanwendungsgebiet der Digitalen Fabrik wird in der ganzheitlichen visuellen und simulativen Betrachtung des gesamten Prozesses der Entstehung zukünftiger Produkte gesehen. Somit wird der Einsatz der Digitalen Fabrik nicht nur auf bestimmte Prozessabschnitte beschränkt. Durch eine ganzheitliche Betrachtung des Entstehungsprozesses sollen Fehler frühzeitig erkannt und eine beschleunigte und parallelisierte Planung ermöglicht werden. Diese Sichtweise definiert die Digitale Fabrik als Bindeglied zwischen Produktentwicklung, Produktionsplanung und Produktion (vgl. [USD11]).

Die Digitale Fabrik kann als lückenloses Abbild aller Elemente und Prozesse einer Fabrik verstanden werden, das in einem Rechenmodell visualisiert wird. In einer solchen *Virtual Reality* wird eine nachvollziehbare Darstellung großer Datenmengen ermöglicht, die zu einer verbesserten Zusammenarbeit unterschiedlicher Bereiche führt (vgl. [Wie01]).

Durch die Visualisierung und Simulation der Strukturen und Fertigungsprozesse einer realen Fabrik wird diese digital *erlebbar*. Das digitale Modell der realen Fabrik kann zur Planung und Optimierung der bestehenden Strukturen verwendet werden. Einen wichtigen Gegenstand der Digitalen Fabrik stellen unterschiedliche Softwarewerkzeuge dar. In einer zentralen Datenbasis sind alle relevanten Informationen

hinterlegt, die für die unterschiedlichen Softwarewerkzeuge genutzt werden. Eine Hauptaufgabe bei der Einführung der Digitalen Fabrik liegt im Aufbau dieser gemeinsamen Datenbasis, in der Nutzung der Softwarewerkzeuge und in der Anpassung der Aufbau- und Ablauforganisation (vgl. [WBK03]). Ein wichtiges Merkmal ist die Vernetzung digitaler Modelle, Methoden und Werkzeuge, die für eine durchgängige Unterstützung der Fabrikplanung und des Fabrikbetriebes notwendig sind (vgl. [Wen04]).

Unter dem Begriff der Digitalen Fabrik werden alle rechnergestützten Werkzeuge für die Entwicklung im gesamten Produktentstehungsprozess zusammengefasst. Jedoch wird zwischen der Planung der Methoden, der Fabrik und der Logistik differenziert. Innerhalb dieser Planungsbereiche verschiedene Planungsebenen definiert (Prozess, Zelle, Segment, Werk) und der Integrationsaspekt der Planungswerkzeuge hervorgehoben (vgl. [Mar06a]).

Mit der VDI-Richtlinie 4499 wird eine einheitliche Regelung für die Umsetzung der Digitalen Fabrik veröffentlicht. Der Begriff der Digitalen Fabrik wird hier wie folgt definiert:

„Die Digitale Fabrik ist der Oberbegriff für ein umfassendes Netzwerk von digitalen Modellen, Methoden und Werkzeugen – u.a. der Simulation und dreidimensionalen Visualisierung – die durch ein durchgängiges Datenmanagement integriert werden. Ihr Ziel ist die ganzheitliche Planung, Evaluierung und laufende Verbesserung aller wesentlichen Strukturen, Prozesse und Ressourcen der realen Fabrik in Verbindung mit dem Produkt“ (vgl. [VDI08]).

Die VDI-Richtlinie 4499 weist zudem darauf hin, dass die eingesetzten und miteinander abgestimmten Methoden und Prozesse genauso wichtig sind, wie die zum Einsatz kommenden Softwarelösungen.

Der Ursprung des Gedankens der Digitalen Fabrik liegt in der kontinuierlichen Entwicklung einer Computertechnologie, die Aufgaben in allen Teilbereichen, wie z.B. der Entwicklung, Fertigung, Montage und Produktionsplanung und -steuerung, eines Industrieunternehmens durch eine digitale Datenverarbeitung unterstützt (vgl. [Sch08]). Durch diese Rechnerunterstützung können Tätigkeiten vereinfacht und ihre Effizienz gesteigert werden. (vgl. [SBLD98]) Beispielhaft wird

dies an der Einführung von Textverarbeitungssystemen und CAD-Systemen (Computer Aided Design (CAD)) im Folgenden verdeutlicht.

Vor der Einführung von Textverarbeitungssystemen zwischen den 1970er und 1980er Jahren hatten einfache Textkorrekturen die Folge, dass der vollständige Text neu verfasst werden musste. Dieser Aufwand wird bei elektronischen Textverarbeitungssystemen durch eine interaktive Benutzerschnittstelle stark verringert. Die Einführung hat zur Folge, dass der Zeitaufwand für das Verfassen von Texten abnimmt und die Qualität der so erstellten Dokumente zunimmt. Die schnelle Verbreitung der Systeme sorgte seit dieser Zeit für eine vollständige Verdrängung der Schreibmaschine aus den Büroräumen, obwohl in den Anfangsjahren der wirtschaftliche Nutzen und die Nachhaltigkeit der Textverarbeitungssysteme nur teilweise erkannt wurde (vgl. [Sch08]).

Die Weiterentwicklung der Funktionen bezüglich interaktiver Zusammenarbeit in Form von Dokumenten-, Wissens- oder Workflowmanagement ist auch zum aktuellen Zeitpunkt nicht abgeschlossen. Die Einführung der interaktiven Textverarbeitungssysteme macht rückblickend erst eine moderne Arbeitsteilung und die damit einhergehende effizientere Produktion möglich (vgl. [USD11]).

Durch die Einführung von CAD-Systemen in den 1980er Jahren können Konstrukteure und Ingenieure ähnliche Vorteile wie bei der elektronischen Textverarbeitung für deutlich komplexere Tätigkeiten nutzen. CAD-Systeme ermöglichen, technische Entwürfe digital auszuarbeiten, zu ändern, wiederzuverwenden und zu ergänzen. Das bis dahin verwendete Radieren hingegen erlaubt bei einer manuellen Bearbeitung auf einem Papier maximal sechs Änderungen an einer Stelle (vgl. [Spi12], [USD11]). Die Einbindung von geometrischen Funktionen vereinfacht Konstruktionsschritte, wie z.B. eine Eingabehilfe für die Erstellung einer Parallelen zu einer Linie. Objekte können im virtuellen Raum gedreht, von allen Seiten betrachtet und ohne größeren Aufwand zerlegt werden. Das parametrische Modellieren ermöglicht das Verknüpfen aller Maße eines Objektes. Dies bedeutet, dass eine Veränderung eines angegebenen Längenmaßes eine Anpassung aller anderen Maße zur Folge hat. Diese Funktionen von CAD-Systemen

zeigen die Vorteile einer virtuellen gegenüber einer manuellen Konstruktionszeichnung, indem Routinearbeiten erleichtert werden und somit die Konstruktionsgeschwindigkeit eines Objekts beschleunigt wird (vgl. [Sch08]).

Aufgrund des anhaltenden Fortschritts in der Informationstechnik wird mit dem CIM-Konzept (Computer Integrated Manufacturing (CIM)) bereits seit Anfang der 1980er Jahre versucht, eine Integration der verschiedenen Unterstützungssysteme umzusetzen (vgl. [Sch08]). Das Ziel des CIM-Konzeptes ist die Schaffung einer durchgängigen und lückenlos rechnerunterstützten Kette von Unterstützungssystemen von der Planung bis zur Produktion. Es wird übergreifendes Informationssystem vorgeschlagen, auf das die computergestützten Bereiche wie Entwicklung, Arbeitsplanung, Fertigung, Qualitätssicherung und die Produktionsplanung und -steuerung Zugriff haben. Eine Vernetzung der einzelnen Datenverarbeitungssysteme, die sowohl technisch als auch organisatorisch durchgängig umgesetzt werden muss, ist notwendig, um die Ziele des CIM-Konzeptes, Zeit- und Kostenersparnisse bei der Produktentstehung, zu erreichen (vgl. [Kra90], [Sch90]). Der wesentliche Gedanke bei dieser Entwicklung liegt in der Zielsetzung einer vollautomatischen Fertigung. Er erfordert tiefgreifende personelle und organisatorische Voraussetzungen innerhalb der Geschäftsprozessketten (vgl. [Gad12], [Sch99]).

Drei grundsätzliche technische Herausforderungen begleiten die bisherigen Umsetzungsversuche der Digitalen Fabrik:

- Eine unerwartet große Schnittstellenkomplexität zwischen isolierten Softwarewerkzeugen und -systemen, die Handhabung unterschiedlicher Programmiersprachen und die erforderliche Datenspeicherkapazität.
- Die Inflexibilität starrer, vollautomatischer Fertigungslinien, die bei einer Umstellungen auf ein neues Produkt zu hohen Kosten führt, so dass eine Massenproduktion von zunehmenden Produktvariationen nicht rentabel durchgeführt werden kann.
- Modellierung und permanente Synchronisierung des digitalen Abbilds der realen Fabrik.

2.2 Industrie 4.0

Das Konzept der Digitalen Fabrik bezieht sich auf die Abbildung einer realen Fabrik und ihrer Produkte, jedoch nicht auf die Ausgestaltung der Produktionsmethoden selbst. Dies bedeutet, dass zukünftige Anforderungen an die Produktions- und Logistiksysteme, wie z.B. eine immer stärker zunehmende Individualisierung der Produkte, kürzere Produktlebenszyklen und eine stetig steigende globale Konkurrenz, nicht alleine über das Konzept der Digitalen Fabrik zu lösen sind (vgl. [Sen13]).

In einer Wandlung hin zu einer *Industrie 4.0* wird die Digitale Fabrik in die reale Fabrik integriert, es entsteht eine *Smart Factory*. In dieser kommunizieren Produkte, Maschinen und Ressourcen direkt miteinander. Es entsteht ein Produktionsumfeld, das aus intelligenten, sich selbst steuernden Objekten besteht. In diesen Cyber-Physischen Systemen (CPS) wird die Vernetzung der dezentralen intelligenten Objekte durch die flächendeckende und wirtschaftliche Verfügbarkeit der technischen Infrastruktur in Form von industriell einsetzbaren Funk- und Internetverbindungen ermöglicht (vgl. [SGG⁺13]). Durch einen flächendeckenden Einsatz in der gesamten Produktionskette entsteht ein sich selbst organisierendes Netzwerk von Maschinen, Lagersystemen und Betriebsmitteln, die untereinander Informationen in Echtzeit austauschen: das *Internet der Dinge und Dienste*. Ein Grundgedanke ist die Nutzung von Fortschritten in der Vernetzungstechnologie. Durch diese Maßnahme wird das Produkt bzw. der Ladungsträger ein aktives Element des Produktions- oder Logistikprozesses (vgl. [Rus13]).

Die deutsche Regierung hat eine *Industrie 4.0* im Zuge ihrer Hightech-Strategie zu einem Kernelement bestimmt. Sie ermöglicht im internationalen Wettbewerb ein schnelles Handeln von Industrie, Wissenschaft, Politik, Gewerkschaften und Gesellschaft, um Wettbewerbs- und Standortvorteile durch den Einsatz der neuen Technologien zu generieren (vgl. [KWH12]). Die Priorität der Umsetzung einer *Industrie 4.0* unterstreicht die Forschungsunion für Wirtschaft und Wissenschaft, die von 2006 bis 2013 als das zentrale innovationspolitische Beratungsgremium die Umsetzung und Weiterentwicklung der Hightech-Strategie in Deutschland begleitet (vgl. [For14]). Laut ihr ist es für Deutschland nur möglich ein erfolgreicher Produktionsstandort

zu bleiben, wenn es gelingt, die Vision der vierten industriellen Revolution mit zu gestalten. Eine zentrale Herausforderung ist die Entwicklung, Vermarktung und der Einsatz von autonomen, selbststeuernden, wissensbasierten und sensorgestützten Produktionssystemen (vgl. [SGG⁺13]). Anhand dieser zu erwartenden Veränderungen, die durch die Einführung der Industrie 4.0 und ihrer neuen Fertigungsverfahren entstehen, prognostiziert die Deutsche Akademie der Technikwissenschaften (Acatech) eine Steigerung der Produktivität von 30% bis 50% (vgl. [Rus13]).

Die Produkte halten die Informationen über ihren Herstellungsprozess und künftigen Einsatzort. Sie sind im gesamten Fertigungsprozess dauerhaft identifizierbar und lokalisierbar und bestimmen den Materialfluss. In Kollaboration mit den anderen Teilsystemen des CPS können so zulässige Szenarien entwickelt und diese unter Berücksichtigung von Optimierungskriterien ausgewählt werden. Hierbei werden der aktuelle Fertigungsstand und alternative Prozessabläufe berücksichtigt (vgl. [Rus13]). Eine dezentrale Materialflusssteuerung ist daher Kernbestandteil des Konzepts einer *Industrie 4.0*.

Zukünftige Intralogistikprojekte werden durch die Vernetzung vieler Subsysteme im Kontext einer Industrie 4.0 einen immer heterogenen Charakter erhalten. Es werden durch die Einführung der intelligenten Objekte mehr Schnittstellen entstehen, die entwickelt und verwaltet werden müssen. Die zu planenden System werden erheblich komplexer. Zudem wird die Transparenz innerhalb der gesamten Produktionskette durch die Verfügbarkeit der in Echtzeit anfallenden Daten steigen. Dies erfordert ein effektives und durchgängiges *Collaborative Engineering* über den gesamten Lebenszyklus eines Gesamtsystems, an dessen Planung, Realisierung und Betrieb eine Vielzahl von Akteuren beteiligt sind.

2.3 Collaborative Engineering

Collaborative Engineering (auch *Kollaboratives Engineering*) bezeichnet die Zusammenarbeit verschiedener Unternehmen (Hersteller, Lieferanten, Kunden, Dienstleister) in verteilten Netzwerken mit unterschiedlicher IT-Infrastruktur bei Entwicklungs-, Planungs-, und/oder Realisierungsprojekten. Meist wird dieser Prozess durch Internettechnologien unterstützt. Um das Projektziel einer unternehmensübergrei-

fenden Planung und Realisierung komplexer, dezentral gesteuerter In-
tralogistiksysteme zu erreichen, gilt es, eben diese effektive und rei-
nungslose Zusammenarbeit zwischen Unternehmen durch an ein kol-
laboratives Engineering angepasste Prozessabläufe zu ermöglichen.

Collaborative Engineering kann wie folgt definiert werden:

“Collaborative Engineering wird als gemeinsames Bearbeiten einer Aufgabe durch verteilte Akteure bezeichnet. Es ist eine Form der unternehmensübergreifenden integrierten Zusammenarbeit über den gesamten Lebenszyklus eines Produktes oder einer Technologie, wobei der Schwerpunkt der Aktivitäten im Bereich der Produktentwicklung liegt. Collaborative Engineering zielt auf die gemeinsame Erarbeitung technologischer bzw. wettbewerblicher Vorteile, die durch einzelne Akteure allein nicht erreichbar wären. Die Realisierung von Collaborative Engineering bedingt u.a. intensive Kommunikation zwischen den Partnern und geeignete Organisationsformen zur Steuerung überbetrieblicher Arbeitsgruppen. Informationstechnische Konzepte wie das Internet und begleitende personalwirtschaftliche Aktivitäten wie Coaching können ergänzend zum Einsatz gebracht werden, um die Effektivität des Collaborative Engineering-Prozesses zu erhöhen” (vgl. [Wir11]).

Weitere ähnliche Definitionen des Begriffes finden sich in [CM07], [FL05], [WSX⁺02], [MS02] und [KN08].

Als *Collaboration* wird die Zusammenarbeit zwischen Partnern verstanden, deren Geschäftsprozesse eng miteinander verzahnt sind. Im Rahmen des *Collaborative Engineering* arbeiten mehrere geografisch verteilte Standorte an einem gemeinsamen Projekt, wozu in geeigneter Weise Wissen und Daten anwendungsgestützt zur Verfügung gestellt und ausgetauscht werden müssen. Mit *Collaborative Engineering* sind wirtschaftlich relevante Erwartungen verbunden: niedrigere Kosten, kürzere Bearbeitungszeiten, Einhaltung der vereinbarten Qualität im Rahmen der Abwicklung von Aufträgen sowie Erhöhung der Reaktions- und Innovationsfähigkeit, stärkere Kundenorientierung und -bindung (vgl. [RBGH06]).

Folgende Probleme können beim kollaborativen Engineering auftreten und sind durch kollaborative Engineeringprozesse zu vermeiden (vgl. [iFA11]):

- Speicherung von Informationen in unterschiedlichen Systemen/an unterschiedlichen Orten
- Verwendung verschiedener/inkompatibler Formate
- Fehlender Zugang zu existierenden Informationen
- Parallele Bearbeitung verschiedener Dokumentversionen

Der Erfolg des Konzepts ist von der Effektivität der oben angesprochenen Verteilung, der gezielten Auslagerung von Teilaufgaben im Rahmen der Supply Chain abhängig. Die einzelnen Teilaufgaben müssen dabei sicher überwacht werden und die jeweils erbrachten Leistungen und deren Ergebnisse zusammengefügt werden, um die vertraglich garantierte Gesamtleistung zu erreichen (vgl. [RBGH06]).

Arbeitsteilige Engineeringprozesse in räumlich getrennten Teams werden im Rahmen der Globalisierung und Unternehmensvernetzung verstärkt erforderlich. Mit den vielfältigen Kommunikationsmöglichkeiten des Internets kann diesen Anforderungen begegnet werden (vgl. [CTG96]). Prozesse des *Collaborative Engineering* werden durch Projektmanagementsoftware unterstützt (vgl. [Co.11]).

Relevante Grundelemente sind IT-Systeme bzw. Softwarewerkzeuge, wie z. B. CAD- und CAM-Werkzeuge, die eine Integration der verteilten Teams ermöglichen. Neben solchen *Integration Technologies* sind Werkzeuge der gemeinsamen Planung und des Projektmanagements notwendig (vgl. [Ker05]).

Das *Collaborative Engineering* wird umfassend durch die von Internettechnologien unterstützte Gestaltung verteilter, gemeinschaftlicher Produktentwicklungsprozesse aus technologischer, organisatorischer und personeller Sicht bestimmt (vgl. [KK02]). Die Entwicklungspartner stammen dabei entweder aus unterschiedlichen Bereichen bzw. Standorten eines Unternehmens oder aus einem unternehmensübergreifenden Netzwerk, das auch Kunden umfassen kann.

Als wesentliche Gestaltungsdimensionen werden die technologische, die organisatorische und die personelle Dimension identifiziert. Im

Kontext eines unternehmensübergreifenden *Collaborative Engineering* ist die Formulierung einer Kooperationsstrategie erforderlich (vgl. [KK02]).

Das Konzept *Collaborative Engineering* kann als eine Weiterentwicklung des *Concurrent Engineerings* betrachtet werden, welches ein Ansatz zur Überwindung der strikten Arbeitsteilung in Produktentwicklungsprozessen darstellt. Dabei wird das Adjektiv *Concurrent* zur Bezeichnung zeitparalleler Vorgänge eingesetzt, wobei es neben Parallelisierung von Prozessschritten einen wesentlichen Fokus auf die interdisziplinäre Zusammenarbeit als Kernelement einer erfolgreichen Produktentwicklung legt (vgl. [SK97]).

Die Gartner Group definiert den Begriff *Collaborative Engineering* als Mittel zur globalen gemeinsamen Nutzung von Engineering Ressourcen durch mehrere Abteilungen unter Einbeziehung von Kunden und Lieferanten bei der Schaffung von neuen Produkten (vgl. [Ree01]). Eine effektive Zusammenarbeit kann nur durch eine digitale Unterstützung des gesamten Entwicklungsprozesses gewährleistet werden, indem alle für die Ausführung des aktuellen Entwicklungsschrittes erforderlichen Daten und Informationen sowie die entsprechenden Anwendungssysteme zur Verfügung stehen (vgl. [BSWW09]). Der Fokus der Kommunikation liegt auf der Art und Weise, wie Teilnehmer in Kooperationen miteinander kommunizieren. Dieser Aspekt umfasst die beteiligten Nutzer und ihre informationstechnischen Fähigkeiten, die Kommunikationskanäle, die dazu verwendeten Anwendungen und die ausgetauschten Informationen.

Bestehende Anwendungen beruhen darauf, dass Informationen oder Anwendungen zur gemeinsamen Verwendung im Rahmen der jeweiligen Zusammenarbeit auf einer gemeinsamen Plattform abgelegt werden, die entweder von einem der Partner aufgebaut oder von einem Dienstleister betrieben und zur Verfügung gestellt wird. Der Zugriff auf diese gemeinsame Plattform ist zwischen den Beteiligten zu koordinieren. Es werden Dokumente, Spreadsheets oder komplexere Anwendungen zum Beispiel zur Datenerfassung oder technischen Auslegung gemeinsam genutzt. Zusätzlich enthalten diese Plattformen typische Collaboration-Hilfsmittel, wie etwa Projektkalender oder workflowunterstützende Funktionalitäten (vgl. [Hay09]).

Jedoch eignen sich diese existierenden Lösungen nicht für ein *Collaborative Engineering* im vorliegenden Forschungsprojekt adressierten Anwendungsfall der Planung und Inbetriebnahme heterogener, dezentral gesteuerter Materialflusssysteme. Für diesen innovativen Anwendungsfall wird eine spezielle Software entwickelt und getestet, die eine effiziente Bearbeitung von heterogenen Großprojekten im Bereich der Materialflusstechnik unterstützt.

Im folgenden Abschnitt wird zunächst der Stand der Forschung aus Sicht der logistischen Anwendungsfälle beschrieben, um im zweiten Teil aktuelle Entwicklungen im Bereich der kollaborativen Softwareentwicklung wiederzugeben. Ziel ist es hierbei, die exponierte Eignung verteilter Softwareentwicklung für den vorgeschlagenen Einsatzfall in der Logistik aus den technologischen Entwicklungen dieses Bereichs abzuleiten.

2.4 Verteilte und kollaborative Softwareentwicklung

Die Erstellung von Software zur Unterstützung betrieblicher Abläufe wird in zunehmendem Maße komplexer. Da der Erstellungsprozess in der Softwareindustrie traditionell einer Werkstatt- bzw. Einzelfertigung entspricht, erfordert die stetig steigende Nachfrage nach betrieblicher Software und die fortschreitende Globalisierung die rationelle Gestaltung der Softwareentwicklung. In der Literatur werden daher immer häufiger die Industrialisierung der Softwareerstellung und neuartige Formen der Spezialisierung, Arbeitsteilung und Zusammenarbeit vorgestellt. Dabei kann im Wesentlichen unterschieden werden, ob die Zusammenarbeit einzelner Akteure und Arbeitsgruppen auf Projektebene oder die strategische Zusammenarbeit von Unternehmen innerhalb der Softwareindustrie behandelt wird. Über diese beiden grundlegenden Betrachtungsebenen hinweg lassen sich existierende Ansätze zur arbeitsteiligen Softwareerstellung entlang mehrerer Dimensionen, wie die räumliche, zeitliche und organisatorische Verteilung der Aktivitäten im Prozess sowie Intensität und Richtung der Zusammenarbeitsbeziehungen klassifizieren (vgl. [HRH06]).

Die Entwicklung von Software erfordert stets die Zusammenarbeit von mindestens zwei Personen (Auftraggeber und Auftragnehmer). Bei komplexeren, über mehrere Teams oder Organisationen verteilten

Softwareprojekten haben zusätzlich sogenannte Gruppenprozesse erheblichen Einfluss auf den Projekterfolg bzw. -misserfolg. Neben den Prozessen Kooperation, Koordination und Kommunikation sind hierbei auch menschliche Faktoren (Kontext), die durch die umgebende Organisation sowie die persönliche Motivation und Kompetenz geprägt werden, von Bedeutung. Die Beteiligung mehrerer, fachlich sehr unterschiedlicher Akteure führt zu speziellen Anforderungen an die jeweilige Werkzeugunterstützung der relevanten Aktivitäten im Prozess. Es müssen daher sowohl die einzelnen Disziplinen und/oder Aktivitäten mit entsprechenden Werkzeugen individuell unterstützt werden als auch eine zentrale *Kollaborationsplattform* zur Projektkoordination bereit stehen.

Im Open Source-Bereich wird diese Funktion im Wesentlichen von Projektplattformen wie SourceForge bzw. den entsprechenden Komponenten für Versionskontrolle (Subversion, CVS etc.) und Gruppenkommunikation (Mailinglisten, Issue Tracker, Foren etc.) erfüllt. Auch Hersteller von kommerziellen Entwicklungswerkzeugen, wie beispielsweise IBM Rational und Borland, bieten immer mehr Teamfunktionalitäten an und unterstützen damit vermehrt kollaborative Softwareerstellungsprozesse. Die Frage, welche Funktionen für welche Aktivitäten geeignet bzw. essenziell sind, ist bisher hingegen noch weitestgehend ungeklärt (vgl. [RBGH06]).

Zur Unterstützung der kollaborativen, zwischenbetrieblichen Softwareerstellung bedarf es daher geeigneter Methoden und Werkzeuge, um diese unternehmensübergreifenden Entwicklungsprozesse zu realisieren. Insbesondere bei der Methodik bzw. dem Vorgehensmodell stellt sich die Frage, inwieweit existierende Ansätze geeignet sind, um diese neue Form der Zusammenarbeit zu unterstützen. In logistischen Softwareprojekten besteht die Besonderheit, dass in den Projektphasen sehr unterschiedliche Werkzeuge Verwendung finden.

Kollaborative Ansätze in der Softwareentwicklung werden, ähnlich wie die Werkzeuge des Collaborative Engineering, aus anderen Ingenieurdisziplinen verwendet (z.B. Computer Aided bzw. Concurrent Design Prozesse im Automobilbau). Bisher befinden sich die meisten dieser Ansätze in der Softwareentwicklung, u.a. aufgrund der speziellen Eigenschaften von Software, in einem prototypisch-experimentellen Stadium (vgl. [HRH06]). Nebenläufige Entwurfsprozesse werden auch im-

mer stärker in UML-basierte Methoden zur modellgetriebenen Softwareentwicklung (Model-Driven Development (MDD)) integriert. Wenn auch bisher nur wenige fundierte Methodiken existieren, ermöglichen moderne Entwicklungswerkzeuge, wie beispielsweise Poseidon for UML Enterprise (vgl. [Gen11]), Rational XDE (vgl. [IBM11]), nebenläufige, verteilte und modellgetriebene Softwareerstellungsprozesse.

Eine Zusammenarbeit in der Implementierungsphase kann sowohl synchron in Form von Shared Workspaces bzw. Application Sharing (z.B. Collaborative Editing), als auch asynchron, gepuffert über Versionsmanagementsysteme (Code Repositories), unterstützt werden (vgl. [DR93]). Mit der CAIS-Methode (Collaborative Asynchronous Inspection of Software, vgl. [MDTR93, MFR94]) existiert ein verteilter Ansatz zur kollaborativen Qualitätssicherung als Teil der Postimplementierungsphase vgl. [SRHM97]. Ebenso in diese Kategorie fallen Ansätze zur verteilten und asynchronen Wartung bereits entwickelter Software (Collaboration in Software Maintenance, vgl. [LR93]).

Alle genannten Ansätze sind jedoch nur auf softwaretechnischer Ebene angesiedelt und schenken den physikalischen Prozessen in einem Materialflusssystem keine besondere Beachtung. Es fehlen kollaborative Werkzeuge, die ähnliche Funktionen, wie existierende Simulations- und Emulationsprogramme für die Planungs- und Realisierungsphase, bieten, d.h. die auch den physischen Materialfluss nachbilden und visualisieren. Insbesondere für die bereits genannten heterogenen Großprojekte, bei denen mehrere Hersteller ihre unterschiedlichen Systeme koppeln müssen, fehlt eine Unterstützung für virtuelle Kopplungstests und die gemeinsame Definition von Schnittstellen.

Zusammenfassend kann gesagt werden, dass Kosten- und Zeitersparnisse in Projekten mit dezentral gesteuerten Anlagenteilen erst durch eine konsequente Verlegung vieler Arbeiten aus der Inbetriebnahme-phase beim Kunden in vorgelagerte Phasen beim Hersteller realisiert werden können. Dieser Prozess wird durch die bisher verwendeten (Software-)Werkzeuge und Engineeringprozesse nicht ausreichend unterstützt. Dies gilt vor allem, wenn mehrere Hersteller an einem Projekt beteiligt sind.

2.5 Planung und Umsetzung technischer Logistiksysteme

In den folgenden Unterabschnitten wird der Stand der Forschung in *Erstellung, Betrieb und Erweiterung von Materialflusssystemen* beschrieben. Die Abschnitte zeigen Unterschiede im Engineeringprozess zentral und dezentral gesteuerter Anlagen auf, die es im vorliegenden Forschungsprojekt zu berücksichtigen gilt. Der hier beschriebene Projektablauf bildet die Grundlage für die Entwicklung des kollaborativen Engineeringprozesses (vgl. Kapitel 4), der in diesem Forschungsprojekt entwickelt wird. Im Rahmen des Tauglichkeitsnachweises der entwickelten Kollaborationssoftware wurden die Erweiterungen der in den folgende Abschnitten beschriebenen Prozessschritte bei der Koppelung der existierenden Versuchsanlagen erfolgreich durchlaufen werden (vgl. Kapitel 9).

Erstellung, Betrieb und Erweiterung von Materialflusssystemen

Die Prozesse von der Planung bis hin zum Betrieb von Materialflussanlagen lassen sich grob in vier Phasen einteilen (vgl. Abbildung 2.2). In jeder dieser Phasen fallen spezifische Tätigkeiten an, welche auf Herstellerseite unterschiedliche Anforderungen an Technik und Personal stellen. Die Arbeitsinhalte der einzelnen Phasen werden im folgenden Abschnitt kurz vorgestellt, um den Rahmen zu verdeutlichen, in dem sich die vorgestellte Forschungsidee bewegt. Gleichzeitig sollen die in Abschnitt 1 erwähnten Zeitersparnisse und Verschiebungen von Arbeitsinhalten erläutert werden, die mit einem Engineering dezentral gesteuerter Anlagen einhergehen.

Planungsphase

Die Planungsphase wird durch den Kundenwunsch nach einer Neu- oder Umplanung einer fördertechnischen Anlage bzw. eines ganzen Materialflusssystemes angestoßen. Anbieter von Materialflusstechnik erfassen zunächst gemeinsam mit dem Kunden die Anforderungen an das System in Form eines Lastenhefts. Vom Hersteller werden aufbauend auf den Anforderungen erste Lösungskonzepte in einer Studie erarbeitet und auf ihre Machbarkeit überprüft. In Zusammenhang mit

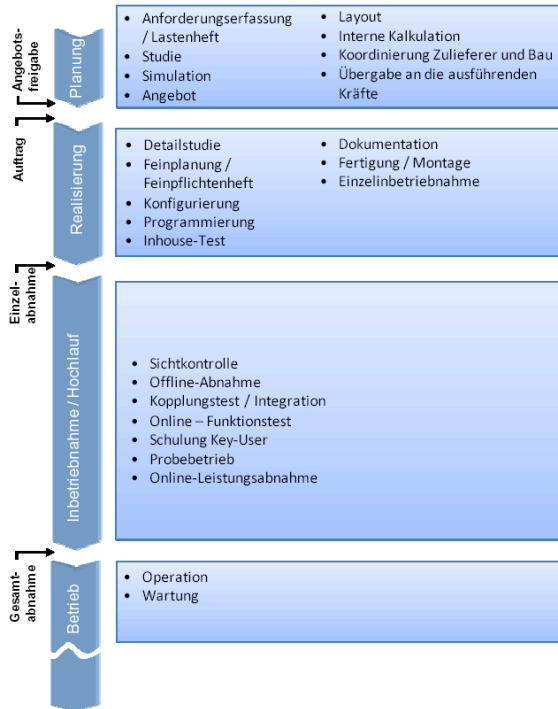


Abbildung 2.2: Tätigkeiten von der Planung bis zum Betrieb einer Materialfluss-anlage und die wichtigsten Meilensteine (Quelle: Forschungsprojekt *Internet der Dinge*) ([GHM10, S.175])

diesem Arbeitsschritt können bei größeren Projekten zudem Simulationsmodelle zum Einsatz kommen, welche eine Abschätzung der prinzipiellen Tauglichkeit der Lösungskonzepte unterstützen. Die aus Studie und ggf. Simulation gewonnenen Erkenntnisse fließen anschließend in eine Bewertung der verschiedenen Konzepte ein. Ein zusätzliches Auswahlkriterium stellt eine erste Kostenrechnung für die betrachteten Lösungskonzepte dar, welche die Grundlage für das spätere Angebot an den Kunden bildet. Bei Lösungskonzepten, die nach der Bewertung in die engere Auswahl kommen, ist in einem nächsten Schritt die Eigenfertigungstiefe festzulegen, d.h. für jedes Teilsystem wird eine Make-or-Buy-Entscheidung getroffen. Zulieferer, von denen Komponenten zugekauft werden sollen, sind an den Gesprächen zu beteiligen und ergänzen die Kostenschätzung für das Gesamtsystem. Durch eine erste Projektierung der benötigten Steuerungstechnik und IT erfolgt eine weitere Verfeinerung des Angebots. Die Güte dieser Aufwandsabschätzung ist stark davon abhängig, in welchem Umfang Lösungen verwendet werden (können), welche in bereits realisierten Projekten zum Einsatz kamen. In diesem Fall lassen sich umfangreiche Zusatzinformationen (z.B. Dokumentationen zur einzusetzenden Mechanik, Steuerungstechnik und Verkabelung) wiederverwenden (vgl. [Kuz10, S.18-28]).

Für Anlagen, die nach dezentralen Steuerungsgrundsätzen konzeptioniert und entwickelt wurden, lassen sich in der Planungsphase verschiedene Arbeitsinhalte vereinfachen und somit Bearbeitungszeiten verkürzen (vgl. Abbildung 2.3). Der Grund dafür liegt in einem hohen Wiederverwendungsgrad einmal entwickelter Software- und Technikmodule. Es kann bereits auf umfangreiche Informationen, Erfahrungswerte und evtl. bereits vorhandene Simulationsmodelle für einzelne Module zurückgegriffen werden. Große Simulationsmodelle lassen sich daher mit vergleichsweise geringem Aufwand erstellen. Dadurch, dass technische Details der Module frühzeitig bekannt sind, verringern sich zusätzlich die Aufwände für Kalkulation und Angebotserstellung (vgl. [Kuz10, S.153-160]).

Realisierungsphase

Die Realisierungsphase folgt auf die Planungsphase, falls der Kunde auf das Angebot hin einen Auftrag an den Anbieter erteilt. In einer Detailstudie werden nun auch spezielle Materialflusstrategien (z.B. Auf-

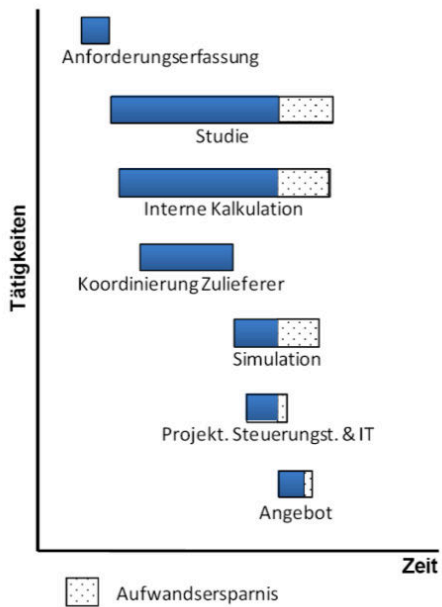


Abbildung 2.3: Erwartete Veränderung des Zeitaufwands in der Planungsphase (qualitativ) ([Kuz10, S. 153])

tragsdisposition, Wegplanung, Betriebsstrategien von Regalbediengeräten) entwickelt und überprüft. Mittels Simulationsmodellen erfolgt eine Leistungsüberprüfung des geplanten Materialflusssystems (z.B. hinsichtlich Durchsatz). Bei der Realisierung kleiner, wenig komplexer Anlagen kann auf Simulationsläufe verzichtet werden. Zudem werden das Layout im Rahmen eines Feinlayoutplans sowie die Systemarchitektur genauer ausgearbeitet und detailliert. Steuerungsaufgaben werden auf die Ebenen der klassischen Automatisierungspyramide verteilt und die Systemhardware (Leitrechner, Materialflussrechner, Automatisierungsgeräte) wird ausgewählt. Anschließend erfolgt die Definition der Systemschnittstellen zwischen den verschiedenen Steuerungsebenen sowie zu übergeordneten Systemen (vgl. [Kuz10, S.18-28]).

Die nächsten Arbeitsschritte liegen in der Fertigung der Fördertechnikkomponenten und der gleichzeitigen Programmierung der Steuerungen. Trotz vorgefertigter Bausteine für die Steuerungsprogrammierung fließt in dieser Phase ein teils erheblicher Aufwand in das so genannte *Customizing*, d.h. in umfangreiche kundenspezifische Änderungen und Anpassungen. Es folgt die Implementierung einer Visualisierungsumgebung, welche sowohl der Überwachung aktueller Systemzustände als auch dem Eingriff in den aktuellen Betrieb dient. Inhouse-Tests beim Softwareersteller dienen der Überprüfung der Steuerungsprogramme auf Materialflussrechner- und SPS-Ebene hinsichtlich Fehlerfreiheit und Funktionsweise im Gesamtzusammenspiel. Parallel zu den Arbeitsschritten *Programmierung* und *Softwaretest* kann bereits mit der Montage der Anlage beim Kunden begonnen werden (vgl. [Kuz10, S.18-28]).

Dezentrale Steuerungskonzepte weisen auch in der Realisierungsphase positive Effekte auf (vgl. Abbildung 2.4). In der Detailstudie kann aufgrund der konsequenten Standardisierung, vor allem im Bereich der Schnittstellen, davon ausgegangen werden, dass sich der Anteil der projektspezifischen Softwarebausteine verringert. Für die Simulation entwickelte Steuerungsprogramme lassen sich direkt auf die reale Anlage übertragen. Somit verringert sich der Programmieraufwand. Ein Modulbaukasten kann spezielle Entitäten für bestimmte Sicherheits-, Kunden- und Werksanforderungen bevorraten und somit den Anpassungsaufwand von Soft- und Hardware für konkrete Kundenlösungen (*Customizing*) zusätzlich verringern. Dezentral gesteuerte Systeme ermöglichen darüber hinaus den schrittweisen Aufbau und Test von

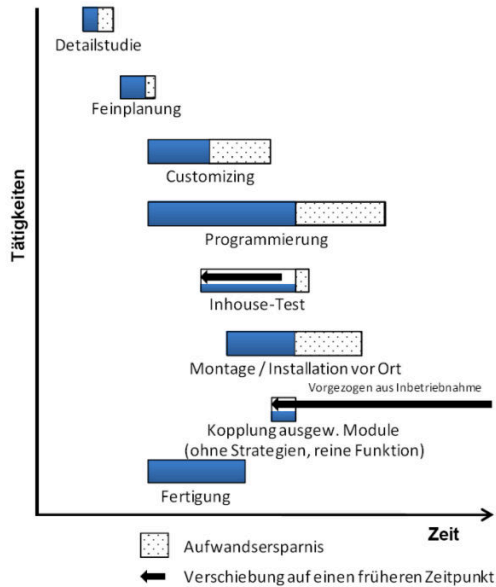


Abbildung 2.4: Erwartete Veränderung des Zeitaufwands in der Realisierungsphase (qualitativ) ([Kuz10, S. 155])

Materialflusssystemen. Dadurch kann zu einem früheren Zeitpunkt als bisher mit Inhouse-Tests begonnen werden. Auch der Kopplungstest beim Kunden wird durch die hohe Anzahl verwendeter Standardkomponenten und damit Standardschnittstellen vereinfacht und kann schon während der Realisierungsphase beginnen, sobald erste Teilsysteme aufgebaut sind (vgl. [Kuz10, S.153-160]).

Inbetriebnahme-/Hochlaufphase

Gewöhnlich findet der angesprochene Kopplungstest erst in der Inbetriebnahme - /Hochlaufphase, die aus mehreren aufeinander aufbauenden Test- und Abnahmeprozessen besteht, statt. Zu Beginn dieser Phase ist das Materialflusssystem mechanisch aufgebaut, die Verkabelung durchgeführt und die Steuerungssoftware eingespielt. Treten zu diesem Zeitpunkt noch Fehler auf, so muss umgehend durch Experten vor Ort nachgebessert werden. Die Reihe der Tests beginnt

mit einer Sichtkontrolle hinsichtlich der handwerklichen Qualität von mechanischem Aufbau und Verkabelung sowie der statischen Eigenschaften der Fördertechnik. Anschließend werden einzelne Anlagenteile in Bewegung versetzt und auf korrekte Funktionalität überprüft. In der Offline-Abnahme werden Handbetrieb- und Halbautomatikfunktionen getestet. Der Kopplungstest dient der Überprüfung eines fehlerfreien Nachrichtenaustausches innerhalb der Anlage. Im Online-Funktionstest werden schließlich detaillierte Testszenarien gefahren. Fehler können in dieser Projektphase meist nur noch sehr aufwändig durch Eingriffe sowohl in die Programmierung des Materialflussrechners als auch in die der Speicherprogrammierbaren Steuerungen (SPS) behoben werden. Während dieser letzten Tests beginnt bereits die Ausbildung sogenannter Key-User, welche ihr Wissen an die restlichen Anlagenbediener weitergeben sollen. Mit dieser Schulung findet meist auch der Gefahrenübergang vom Anlagenhersteller auf den Kunden statt (vgl. [Kuz10, S.18-28]).

Mit der Vielzahl an durchzuführenden Tests und dem hohen Integrationsrisiko stellt die Inbetriebnahme-/Hochlaufphase für den Anlagenhersteller meist die aufwändigste Projektphase dar. Auewech hier versprechen dezentrale Steuerungskonzepte eine Reduzierung von Arbeitsumfängen und Risiken (vgl. Abbildung 2.5). So können Einzelbetriebnahme, Offline-Abnahme und Kopplungstest genauso wie zuvor der Anlagenaufbau parallel für die verschiedenen Anlagenbereiche durchgeführt werden. Hersteller mechatronischer Module nehmen diese bereits *inhouse* vor der Auslieferung in Betrieb, so dass Aufwand und Fehlerrisiko vor Ort sinken. Dies begünstigt auch einen früheren Startzeitpunkt für den Probetrieb einzelner Anlagenteile und schließlich auch für die Gesamtanlage. Da die Steuerungsprogramme auch in einer Simulation verwendet werden können und die Module schon bei Auslieferung mit einer eigenen Visualisierungsumgebung ausgestattet sind, steht bereits zu Beginn der Inbetriebnahme-/Hochlaufphase ein detailliertes Simulationsmodell inkl. Visualisierung für Schulungszwecke zur Verfügung. Daher kann auch mit der Key-User-Schulung zu einem früheren Zeitpunkt als bisher begonnen werden (vgl. [Kuz10, S.153-160]).

Zu einem erhöhten Aufwand kann die dezentrale Steuerungsarchitektur allerdings bei den Arbeitsschritten *Online-Funktionstest* und *Online-Leistungsabnahme* führen. Der Mehraufwand kann bei der Feh-

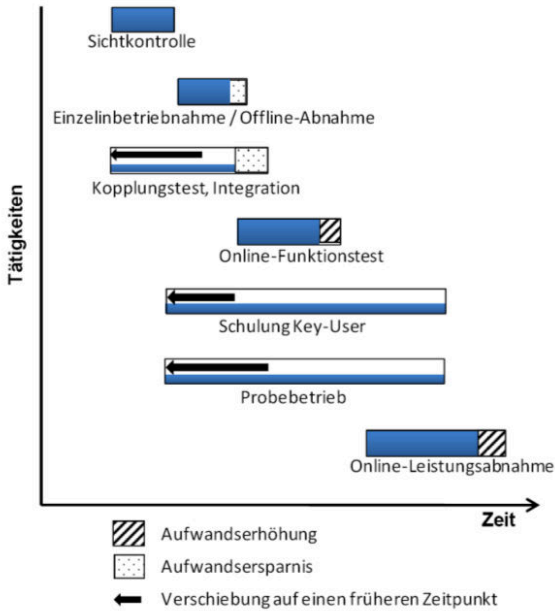


Abbildung 2.5: Erwartete Veränderung des Zeitaufwands in der Inbetriebnahme- /Hochlaufphase (qualitativ) ([Kuz10, S. 157])

lersuche und -behebung entstehen, da Fehler in der Strategieimplementierung und in der Kommunikation in einem dezentralen System schwerer nachzuvollziehen sind als in einem vergleichbaren hierarchisch aufgebauten System. Diese negativen Auswirkungen können durch gezielte Simulation in der Realisierungsphase sowie durch den Einsatz spezieller Softwaretools zur Fehlersuche und -behebung in verteilten Systemen eingegrenzt werden (vgl. [Kuz10, S.153-160]).

Betrieb

In der Betriebsphase sind für den Anlagenersteller vor allem effiziente Möglichkeiten zur Ferndiagnose und Fernwartung des installierten Systems von großer Bedeutung. Zum einen müssen im Rahmen der Gewährleistungspflicht noch die vom Ersteller zu verantwortenden Män-

gel beseitigt werden. Darüber hinaus übernimmt der Anlagenersteller nach der Inbetriebnahme häufig die Rolle des Service-Dienstleisters. Die Eingrenzung von auftretenden Fehlern via Ferndiagnose unterstützt eine gezielte Suche und Behebung durch Personal vor Ort und kann ggf. den Einsatz eigener Service-Mitarbeiter vermeiden. Kommt es trotz festgelegter Wartungsintervalle zum Ausfall von Anlagen, so müssen für Wartungs- und Reparaturarbeiten oftmals Teile oder die komplette Anlage stillgelegt werden. Dies ist mit Zusatzkosten für das Wartungspersonal verbunden, da derartige Stilllegungen meist nur in betriebsfreien Zeiten (nachts oder an Wochenenden und Feiertagen) möglich sind (vgl. [Kuz10, S. 18-28]).

Der Normalbetrieb einer dezentral gesteuerten Materialflussanlage darf sich nicht von dem herkömmlicher Systeme unterscheiden. Im Falle von unerwarteten Anforderungen oder anstehenden Wartungsarbeiten verhält sich das dezentrale System jedoch anders als eine herkömmliche, zentral gesteuerte Anlagen: Die Fördertechnikmodule sind so gestaltet, dass sie auf neue Anforderungen selbstständig reagieren. Bei der Überlastung oder Störung einer Wegverbindung wird automatisch nach Alternativrouten gesucht. Dies begrenzt die Auswirkungen von Fehlern und Ausfällen und verringert die Wahrscheinlichkeit eines Totalausfalls des Systems. Wartungsarbeiten können im laufenden Betrieb erfolgen, d.h. Module oder Anlagenbereiche können stillgelegt werden, ohne dass das restliche System abgeschaltet werden muss (vgl. [Kuz10, S. 153-160]).

Erweiterung / Modernisierung

Die spezielle Herausforderung an Erweiterungs- bzw. Modernisierungsprojekte ist die mechanische und softwaretechnische Integration neuer Gewerke in bereits bestehende Anlagenteile (vgl. [Kuz10, S. 18-28]). Zeit- und kostenintensiv ist in dieser Phase vor allem die Inbetriebnahme, gefolgt von der Umprogrammierung des Materialflussrechners (vgl. [GCK08]). Bei der Erweiterung von dezentral gesteuerten Materialflusssystemen lässt sich dieser Aufwand reduzieren. Ein Simulationsmodell der Anlage kann auf Basis der Steuerungslogik der bestehenden Module aufwandsarm abgeleitet und um die Steuerungslogik neuer Anlagenteile ergänzt werden. Die Möglichkeit einer Emulation erlaubt es zudem, die Steuerung des neuen Anlagenteils in die bestehende Anlage zu integrieren und sozusagen *online* zu testen. An

der bestehenden Anlagenprogrammierung müssen keine Änderungen vorgenommen werden, da im Internet der Dinge Steuerungsstrategien unabhängig von der Anzahl der beteiligten Module entwickelt werden. Dies erlaubt die Integration neuer Anlagenteile ohne Stilllegung bestehender Anlagenteile. Es ist auch möglich, hierarchisch gesteuerte Anlagen mit dezentral gesteuerten Modulen zu kombinieren bzw. aufzurüsten. Hierzu muss allerdings eine Kommunikationsschnittstelle zwischen altem und neuem System geschaffen werden, beispielsweise in Form eines speziellen Softwaredienstes, der in der Lage ist, Datenpakete beider Seiten zu lesen und weiterzuvermitteln. Mittels einer derartigen Schnittstelle könnten herkömmlich gesteuerte Anlagenteile mit dezentralen Anlagen koexistieren und kooperieren (vgl. [Kuz10, S. 153-160]).

Kuzmany weist allerdings auch auf mögliche Problemfelder eines Mischbetriebs von zentral und dezentral gesteuerten Anlagenteilen hin (vgl. [Kuz10, S. 160]): “Die Entscheidung für eine solche Mischform sollte jedoch gründlich überlegt sein. Die verteilte Steuerung wird ihr Potenzial hier nicht komplett ausspielen können. Außerdem verkompliziert die Kombination beider Konzepte die Anlagensteuerung und benötigt für Aufbau und Wartung Experten, welche sich in beiden Denkweisen zu Hause fühlen”.

Dennoch ist gerade das zuletzt beschriebene Integrationsszenario für das vorliegende Forschungsprojekt von zentraler Bedeutung, da es einen realistischen Anwendungsfall für einen kollaborativen Engineeringprozess darstellt und somit die Anforderungen an die zu entwickelnde Kollaborationsplattform mitbestimmt.

3 Forschungsziel und angestrebte Ergebnisse

Ausgehend vom im vorhergehenden Abschnitt aufgezeigten Stand der Forschung spezifiziert der Abschnitt 3.1 das grundlegende Forschungsziel des vorliegenden Forschungsprojektes und konkretisiert die angestrebten Forschungsergebnisse sowohl methodisch und wirtschaftlich (Abschnitt 3.1.1) als auch hinsichtlich ihres innovativen Beitrags (3.1.2). Um für diese mehrschichtigen Forschungsziele die methodischen Grundlagen zu erarbeiten und in einem Demonstrator umzusetzen, ist ein strukturiertes, kooperatives Vorgehen notwendig, welches sich explizit auch in der Konzeptionierung der Arbeitspakete widerspiegelt (Abschnitt 3.2).

3.1 Forschungsziel

Ziel des Forschungsprojektes ist es, standardisierte kollaborative Engineeringprozesse zu gestalten und zu dokumentieren, die eine unternehmensübergreifende Planung, Realisierung und Inbetriebnahme von komplexen, dezentral gesteuerten Intralogistiksystemen sowie deren Betrieb effizient ermöglichen. Zur Prozessunterstützung soll darüber hinaus eine leistungsfähige, offene Kollaborationsplattform entwickelt, implementiert und iterativ optimiert werden.

3.1.1 Angestrebte Forschungsergebnisse

Im diesem Forschungsprojekt wird eine Empfehlung für kollaborative Engineeringprozesse für dezentral gesteuerte Intralogistiksysteme angestrebt. Besondere Beachtung finden Designvorschriften für die Schnittstellenimplementierung und -verwaltung unterschiedlicher Subsysteme. Um die gemeinsame Projektarbeit zu unterstützen, entstehen Planungs- und Simulationswerkzeuge, die über eine Standardschnittstelle gekoppelt werden können. Komplexe Intralogistiksysteme können somit gemeinsam von mehreren Herstellern/Lieferanten vorausgedacht, getestet und virtuell in Betrieb genommen werden. Abbildung 3.1 zeigt die Einbettung der angestrebten offenen Kollaborationsplattform in den kollaborativen Engineeringprozess. Das strukturierte Vorgehen soll eine effiziente Projektierung heterogener

Fördertechnikanlagen unabhängig von der Größe der beteiligten Unternehmen ermöglichen. Die logistische Funktionalität des Gesamtsystems bleibt dabei unabhängig von den auf Subsystemebene angewandten Steuerungskonzepten (zentral/dezentral) erhalten. Die einzelnen Bestandteile der Plattform und deren Bedeutung für zukünftige Planungs- und Entwicklungsprozesse technischer Logistiksysteme werden im Folgenden kurz beschrieben.

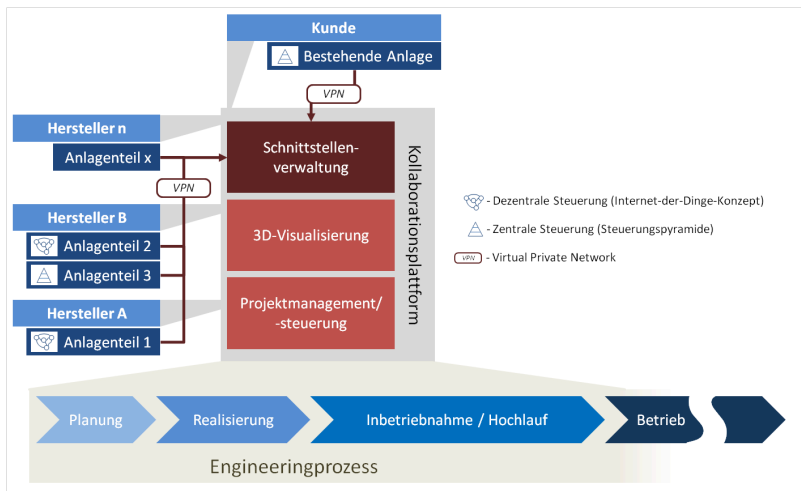


Abbildung 3.1: Kollaborationsplattform als zentrales Element eines herstellerübergreifenden Engineeringprozesses

Die Schnittstellenverwaltung soll, wie oben erwähnt, die Konzeptionierung, den Test sowie die virtuelle Inbetriebnahme komplexer Intra-logistiksysteme ermöglichen, an deren Erstellung mehrere Hersteller beteiligt sind. Dazu muss der Austausch relevanter Informationen über Systemgrenzen hinweg ermöglicht werden. Für die Kommunikation darf es dabei keine Rolle spielen, ob Daten von Steuerungsinstanzen herkömmlicher, hierarchisch strukturierter Anlagen oder direkt von dezentral gesteuerten Modulen gesendet bzw. empfangen werden. Damit wird zum einen gewährleistet, dass für jedes Subsystem eine Wahlfreiheit hinsichtlich des eingesetzten Steuerungsparadigmas besteht. Zum anderen wird der Tatsache Rechnung getragen, dass in der Praxis die zentrale Steuerung fördertechnischer Anlagen über einen

Materialflussrechner weiterhin am stärksten verbreitet ist und dezentral gesteuerte Systeme zum Zeitpunkt der Antragstellung nur vereinzelt angeboten wurden. Daher ist bei der Erweiterung bereits existierender Anlagen damit zu rechnen, dass es sich bei diesen um zentral gesteuerte Systeme handelt. Festzuhalten bleibt allerdings auch, dass sich dezentral gesteuerte Systeme aufgrund ihres stark modularen Aufbaus, des hohen Anteils an wiederverwendbarer Software sowie spezifizierter Kommunikationsschnittstellen besonders gut für die hier angestrebte Form des Engineerings eignen (vgl. Abschnitt 2.3).

Der Austausch von Steuerungs- und Zustandsdaten kann zur Erstellung von Ablaufsimulationen genutzt werden. Ein entsprechendes Testwerkzeug mit 3D-Visualisierung unterstützt den Aufbau der Simulationsmodelle sowie die Durchführung von Testläufen bis hin zur virtuellen Inbetriebnahme. Diese Komponente der Kollaborationsplattform kann dazu mittels Emulation die bestehende Fördertechnik beim Kunden um ein virtuelles Modell der neuen Anlagenteile erweitern. Da die Anlagensteuerung der virtuellen Bereiche bereits auf den später in der realen Anlage eingesetzten Steuerungsprogrammen basiert, lassen sich auf diese Weise erste Tests der Inbetriebnahme-/Hochlaufphase aufwandsarm durchführen. Ein weiteres Ergebnis des Forschungsprojektes stellt eine vordefinierte Vorgehensweise in heterogenen Großprojekten unter Nutzung der Kollaborationsplattform dar. Die entwickelte Vorgehensweise wird dazu mittels geeigneter Methoden modelliert (vgl. Kapitel 4).

Zusammenfassend lässt sich festhalten, dass im vorliegenden Forschungsprojekt Lösungen entwickelt werden, die eine effiziente Projektierung heterogener Fördertechnikanlagen unabhängig von der Größe der beteiligten Unternehmen ermöglichen. Die logistische Funktionalität des Gesamtsystems bleibt dabei unabhängig von den auf Subsystemebene angewandten Steuerungskonzepten (zentral/dezentral) erhalten. Die Schnittstellenverwaltung der Kollaborationsplattform findet nach Projektabschluss auch im operativen Anlagenbetrieb Verwendung und koordiniert die Steuerung der Subsysteme.

3.1.2 Innovativer Beitrag der angestrebten Forschungsergebnisse

Die Entwicklung komplexer, dezentral über einen Schnittstellendienst gesteuerter Materialflusssysteme, die sich aus Anlagenteilen mehrerer

Hersteller zusammensetzen und auf unterschiedlichen Konzepten bzw. Graden von Dezentralität basieren, wird bislang nicht erforscht. Die angestrebten Ergebnisse sollen Kosten und Risiken bei Realisierung, Betrieb und Umbau von heterogenen Materialflusssystemen verringern. Eine erhöhte Adaptivität neuer bzw. erweiterter Systeme sorgt dafür, dass Anpassungen oder Umbauten an den Systemen seltener vorgenommen werden müssen. Werden diese dennoch notwendig, entfällt eine Abschaltung kompletter Anlagen(teile). Dies führt zu einem schnellen, günstigen und risikoarmen Umbau.

KMU sichern ihre Stellung häufig durch ihre, im Vergleich zu Großunternehmen, hohe Innovationsfähigkeit bzgl. neuer Produkte bzw. Systeme und schnelle Reaktionsfähigkeit auf neue Marktbedingungen. Die vorgestellte Projektidee unterstützt diese Eigenschaften. Die angestrebte Standardisierung eines neuartigen Engineeringprozesses vereinfacht die Erschließung neuer Geschäftsfelder für KMU. Zudem beschäftigt sich das Forschungsprojekt erstmals eingehend mit der Gestaltung eines angepassten Planungs- und Integrationsprozesses für eine neue Generation von Materialflusssystemen, welche sich durch dezentrale Steuerungsansätze auszeichnen. Die Praxistauglichkeit derartiger Systeme wird durch einen Abbau von Eintrittsbarrieren und Vorhalten auf Hersteller- und Betreiberseite weiter vorangetrieben.

3.2 Lösungsweg zur Erreichung des Forschungsziels

Um die angestrebten Forschungsergebnisse zu erreichen, war ein interdisziplinärer methodischer Ansatz notwendig, der eine intensive Zusammenarbeit zwischen den Fachgebieten Logistik und Informatik erfordert. Für die Umsetzung wurden insgesamt 8 Arbeitspakete (kurz AP) identifiziert. Im Rahmen der Projektbearbeitung und insbesondere der Softwareentwicklung wurden die Arbeitspakete durch die zwei beteiligten Forschungsstellen weitestgehend parallel bearbeitet. Diese Verfahrensweise wurde bewusst gewählt und steht im Sinne der Ziele des Forschungsprojektes. Diese Zusammenarbeit bezieht sich nicht nur auf die enge Kollaboration bei der Bearbeitung der einzelnen Arbeitspakete, sondern auch auf die Errichtung einer geeigneten technischen Infrastruktur. Der methodische Ansatz zum Erreichen der Forschungsziele bedurfte initial der Definition von grundlegenden Rahmenbedingungen aus Sicht der Logistik – diese wurden im Arbeitspaket *Struk-*

turierung des Problemfeldes und detaillierte Bestimmung der Anforderungen erarbeitet (AP1, Aufwand 6 Personenmonate). Die Ergebnisse wurden in die Analyse gegenwärtiger und zukünftiger Anwendungsfälle der zu realisierenden Kollaborationsplattform umgesetzt. Im Rahmen des Arbeitspaketes „Design kollaborativer Engineeringprozesse (AP2, Aufwand 6 Personenmonate) wurde ein modelliertes Prozessvorgehen für kollaborative Engineeringprozesse gestaltet, welches die Grundlage für die Gestaltung der offenen Kollaborationsplattform bildet. Im Hinblick auf die Gestaltung und Entwicklung einer kollaborativen Softwareplattform wurden in den Arbeitspaketen *Spezifikation und Implementierung der Schnittstellenverwaltung* (AP3, Aufwand 10 Personenmonate), *Spezifikation und Implementierung eines Testwerkzeugs mit 3D-Visualisierung* (AP4, Aufwand 12 Personenmonate) sowie *Entwicklung kollaborativer Softwarewerkzeuge* (AP5, Aufwand 14 Personenmonate) die technischen Voraussetzungen geschaffen um die Zielvorgabe, nämlich ein Softwarewerkzeug zu entwickeln, das ein effektives, abgestimmtes Vorgehen mehrerer Partner bei Planung und Test eines Materialflusssystemes ermöglicht. Dieses Werkzeug stellt eine Plattform dar, die aus zwei Funktionsmodulen besteht. Zum einen aus einem Schnittstellenverwaltungsmodul (AP3) und zum anderen aus einer 3D-Visualisierungsumgebung (AP 4). Beide Module bieten integrierte Funktionalitäten wie Benutzer- und Versionsverwaltung. Deren Funktionalität wurde im Rahmen des Arbeitspaketes *Kopplung existierender Demonstratoren* (AP6, Aufwand 12 Personenmonate) getestet. In diesem AP wurde ein Demonstrationssystem entwickelt, welches aus den separaten dezentralen Steuerungssystemen und Simulationen der beteiligten Lehrstühle in Dortmund und München besteht und dessen Dokumentation als Proof-of-Concept dient. Nach der Fertigstellung der ersten Softwareversion wurden AP2 bis 6 im Sinne eines iterativen Entwicklungsprozesses nochmals durchgelaufen, um die implementierten Softwarewerkzeuge unter Berücksichtigung der Verbesserungsvorschläge aus der Industrie weiterzuentwickeln. Anschließend wurden Maßnahmen getroffen, die zu einer Verbesserung von bereits existierenden Funktionen sowie dem Hinzufügen neuer Funktionen führten. In diesem Zusammenhang wurde der Fokus auf die Robustheit der Software und die Verbesserung der Benutzerfreundlichkeit gerichtet.

Unter Berücksichtigung des Feedbacks wurde innerhalb des Arbeits-

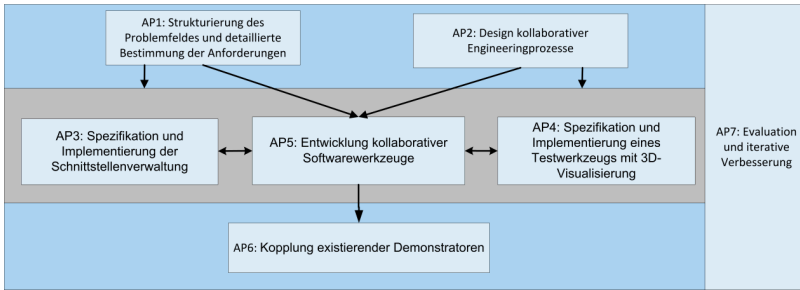


Abbildung 3.2: Methodische Aufteilung des Forschungsprojektes in Arbeitspakete

pakets *Evaluation und iterative Verbesserung* (AP7, Aufwand 2 Personenmonate) ein Plan zur Überarbeitung der existierenden Software entwickelt und dabei wurden die Verbesserungsmöglichkeiten identifiziert und ausgewertet. Die Zwischen- und Endergebnisse wurden im Rahmen des Arbeitspaketes *Dokumentation* (AP8, Aufwand 2 Personenmonate) durch Veröffentlichungen und den vorliegenden Arbeit festgehalten.

Aufbau der Arbeit

Ausgehend von der AP Struktur (siehe Abschnitt 3.2) ist der grundlegende Aufbau dieser Arbeit in Abbildung 3.3 dargestellt. In Kapitel 4 wird ein modelliertes Prozessvorgehen für kollaborative Engineeringprozesse vorgestellt, welches die Grundlage für die Gestaltung der offenen Kollaborationsplattform bildet. Dieses Kapitel dient als Einführung und schafft ein besseres Verständnis für die Ausführungen in dem darauf folgenden Kapitel. In Kapitel 5 wird der Schwerpunkt auf die Strukturierung des Themenfeldes der computergestützten Planung von Intralogistiksystemen gesetzt und außerdem eine Bestimmung der Anforderungen an ein softwarebasiertes Kollaborationssystem vorgenommen. Zu diesem Zweck werden Einsatzszenarien und Anwendungsfälle einer Softwareplattform erarbeitet. In Kapitel 6 werden die Funktionen sowie die Softwarearchitektur der entwickelten KoDeMat-Plattform beschrieben. Besonders im Fokus steht die Verteilung der Daten, da ihre Implementierung ein wesentlicher Teil der

durchgeführten Arbeit ist. Die beiden nächsten Kapiteln (7 und 8) beschreiben die zwei Funktionsmodule der KoDeMat-Plattform, nämlich die Implementierung eines Planungswerkzeugs mit 3D-Visualisierung und der Schnittstellenverwaltung. Anschließend, wird in Kapitel 9 das entwickelte Demonstrationsszenario vorgestellt, welches einen realen Einsatz abbildet.

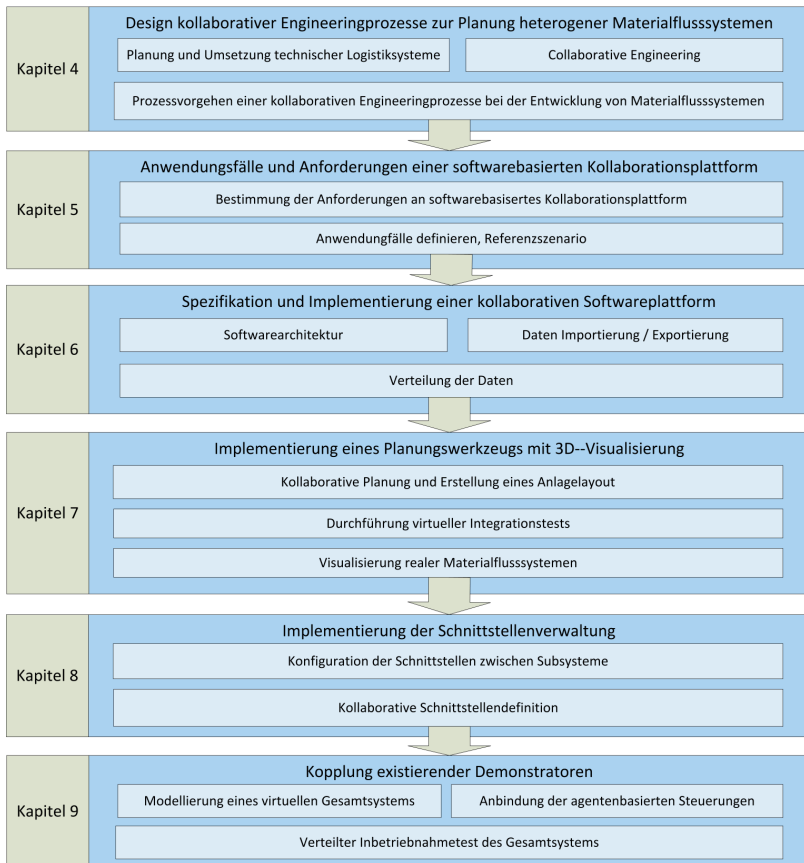


Abbildung 3.3: Aufbau der Arbeit

4 Design eines kollaborativen Engineeringprozesses zur Planung heterogener Materialflusssysteme

Der in diesem Forschungsprojekt entwickelte Kollaborationsprozess wird als Spezialisierung des in Kapitel 2.5 vorgestellten Prozesses für die Planung und Umsetzung technischer Logistiksysteme für dezentrale Materialflusssteuerungen entworfen. Dies bedeutet, dass die grundlegenden Tätigkeiten und Planungsphasen erhalten bleiben und nur die Prozessabläufe genauer spezifiziert werden, welche einen Einfluss auf das Projektziel einer unternehmensübergreifenden Planung und Realisierung komplexer dezentral gesteuerter Intralogistiksysteme haben. Im folgenden werden die Themenbereiche des neuen kollaborativen Planungsprozesses vorgestellt sowie die grundsätzliche Systemarchitektur der dezentralen Intralogistiksysteme dargestellt, welche die Grundlage des Prozesses bilden.

Im Rahmen von Expertengesprächen mit Projektpartnern der Industrie wurden Erfahrungswerte zu den Herausforderungen in den einzelnen Projektphasen gesammelt und insbesondere Problemstellungen bei kollaborativen Projekten betrachtet. Es konnten vier Themenbereiche identifiziert werden, die in einer oder mehr Projektphasen starken Einfluss auf den Projektverlauf nehmen (siehe Abbildung 4.1):

- Planungstätigkeiten
- Konfigurationstätigkeiten
- Tests
- Synchrone Bearbeitung

Unter dem Begriff der Planungstätigkeiten ist in diesem Fall nicht die Planungsphase gemeint, vielmehr fasst er die Tätigkeiten der Planung zusammen, die durchaus auch in anderen Projektphasen weitergeführt werden, wie z.B. ein Update der Layoutplanung. Im Allgemeinen sind diese Planungstätigkeiten Hauptbestandteil der Planungsphase. Zu dem Themenbereich der Konfiguration zählen die Tätigkeiten, welche zur Anpassung eines beteiligten Softwaresystems, d.h.

in der Regel der Materialflusssteuerung, durchgeführt werden müssen. Im Fall von heterogenen Systemen fällt auch die Entwicklung von Schnittstellen zwischen den Subsystemen unter diese Tätigkeiten. Der Themenbereich der Tests umfasst alle Tätigkeiten, die das Zusammenspiel der Subsysteme untereinander und des übergeordneten Gesamtsystems beinhalten. Insbesondere betrifft dies eine frühzeitige Simulation des Gesamtsystems sowie die ordnungsgemäße Funktionalität aller Schnittstellen. Der Themenbereich der synchronen Bearbeitung ist eine technisch-organisatorische Anforderung, die aus der kollaborativen Arbeit entsteht und betrifft die Speicherung der Daten und Dokumente in allen Projektphasen, welche konsistent von allen Beteiligten durchgeführt werden muss.

Um die Bearbeitung dieser Themenbereiche zu unterstützen, kommen heutzutage bereits verschiedene Softwaretools zum Einsatz. Jedoch ist die Mehrheit dieser bestehenden Softwarelösungen allgemein gehalten und somit nicht für den Anwendungsfall der Planung und Inbetriebnahme von Materialflusssystemen optimiert.

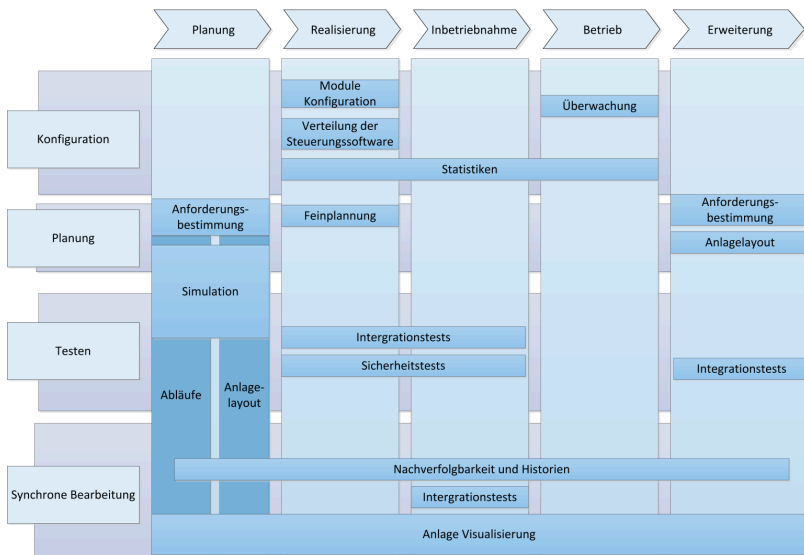


Abbildung 4.1: Tätigkeiten bei der Entstehung einer Materialflusssanlage

Eine effektive und reibungslose Zusammenarbeit zwischen den Projektbeteiligten wird durch einen an ein kollaboratives Engineering angepassten Prozess ermöglicht, der durch ein Softwarewerkzeug unterstützt wird. Der in den folgenden Abschnitten beschriebene Prozess bildet die Grundlage für die Gestaltung der KoDeMat-Plattform und ist in Abbildung 4.2 dargestellt.

Vorarbeiten

Im ersten Schritt des Planungsprozesses findet eine IST-Analyse des aktuellen Zustands der Räumlichkeiten statt. In dieser Phase erfolgt eine Modellierung der räumlichen Gegebenheiten des Gebäudes. In der Regel werden die Daten vom Betreiber (bzw. dem Planer) der bestehenden Anlagen bereitgestellt. Bei Neuanlagen können die entsprechenden 3D-Modelle aus Bauzeichnungen erzeugt werden. Das Ergebnis der IST-Aufnahme ist ein digitales Datenmodell der räumlichen Gegebenheiten des Gebäudes. Die Erfassung des derzeitigen Layouts wird anhand existierender CAD Modelle ermöglicht. Es besteht auch die Möglichkeit mit einem 3D-Laserscanner die aktuellen Installationen innerhalb der Halle zu erfassen und als Modell zu verwenden (vgl. [Kam13], [Vet03]).

Groblayoutplanung - Erstellung von Strukturvarianten

Nachdem ein Abbild der Anlagenteile der Subsysteme innerhalb der Vorarbeiten modelliert wurden, kann in der Grobplanungsphase die Erstellung von Strukturvarianten kollaborativ erfolgen. Das übergeordnete Ziel dieser Groblayoutplanung ist die anforderungsgerechte räumliche Anordnung aller Planungsobjekte (wie z. B. Gebäude, Maschinen, Fördertechnik) und die Detailgestaltung des Hallenlayouts. Bei konventionellen Planungsverfahren werden innerhalb dieses Planungsprozesses gedruckte oder digitale Layoutpläne sowie weitere Dateien zwischen verschiedenen Partnern ausgetauscht und es entstehen somit verschiedene, nicht synchronisierte Versionen der einzelnen Daten. Mit Hilfe internetbasierter Softwarewerkzeuge wie der KoDeMat-Plattform ist es möglich, die oben erwähnten Tätigkeiten des Grobplanungsprozesses durch folgende Funktionen zu unterstützen:

- Eine Synchronisierung von Planungsdaten durch eine enge Einbindung mehrerer verteilter Teams.

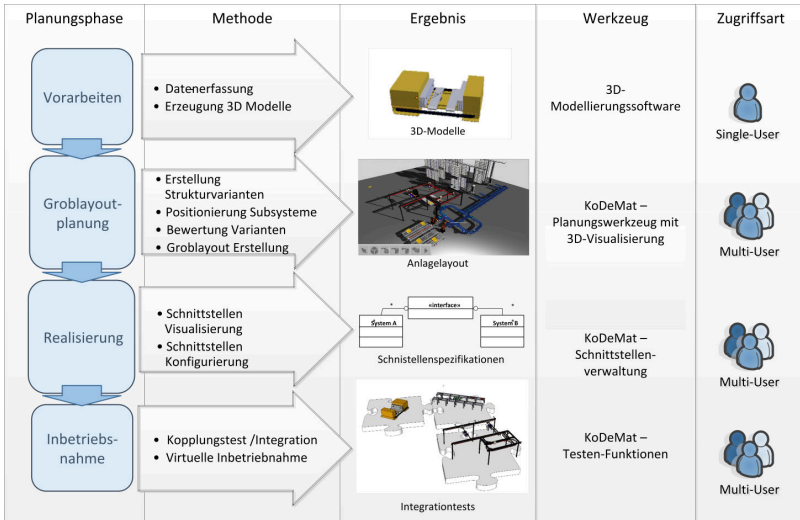


Abbildung 4.2: Planungsprozess eines dezentralen Materialflusssystems mit der KoDeMat-Plattform

- Eine einheitliche 3D-Visualisierung zur Prüfung, Analyse und Zusammenarbeit an Strukturvarianten.
- Eine Nachverfolgung von Änderungen, die es erlaubt, die erzeugten Strukturvarianten zu verwalten, zu genehmigen oder abzulehnen.

Die Realisierung dieser Funktionen durch die Kollaborationsplattform wird im Abschnitt 7.1 detailliert beschrieben.

Realisierung - Systemkonfigurierung

Im dritten Schritt des Planungsverfahrens müssen die heterogenen Subsysteme konfiguriert werden, um miteinander effektiv kommunizieren und zusammenarbeiten zu können. Da der Fokus in dieser Arbeit auf agentengesteuerten Systemen liegt, wird die Konfiguration der Schnittstellen agentenorientiert modelliert (siehe Kapitel 8).

Mittels der im Rahmen des BMBF-Projekts *Internet der Dinge* erarbeiteten Konzepte wird eine vollständig dezentral organisierte Steuerung von Materialflusssystemen ermöglicht. Dazu müssen die Schnittstellen zwischen den dezentral gesteuerten Subsystemen konfiguriert werden. Auch in diesen Systemen existieren zentrale Orte für Systemhersteller sowie -bediener zur Konfiguration des Gesamtsystems. Beispiele hierfür sind die Erstellung der Anlagentopologie, die Festlegung der Arbeitsabläufe sowie die Überwachung des Systems.

Inbetriebnahme – Durchführung von Integrationstests und Visualisierung der Materialflusssysteme

Komplexe Materialflusssysteme müssen für den Betreiber der Anlage und für den Inbetriebnehmer bzgl. ihrer Struktur und ihres aktuellen Zustandes nachvollziehbar sein. Die Visualisierung einer Anlage ist ein wichtiges Hilfsmittel zur Überwachung und zum Monitoring des gesamten Systemverhaltens und ist aus diesem Grund ein unverzichtbarer Bestandteil jedes Materialflusssystemes. Im Fall von Integrationstests von heterogenen, entfernten Systemen ermöglicht eine Multi-User-3D-Visualisierungsumgebung mehreren Benutzern, das Modell eines Materialflusssystemes zeitgleich gemeinsam zu betrachten und potenzielle Fehlerquellen rechtzeitig zu identifizieren.

4.1 Systemarchitektur eines agentengesteuerten Materialflusssystemes

Um die Anwendungsbereiche zu definieren, erfolgt in einem ersten Schritt die Strukturierung von Materialflusssystemen und eine Beschreibung der Systemarchitektur von agentenbasierten Materialflussteuerungen. Die in dieser Arbeit betrachteten Materialflusssysteme können in drei Ebenen strukturiert werden (Abbildung 4.3):

- System: Das betrachtete Gesamtsystem einer Materialflussanlage. Dieses kann weiter in Subsysteme unterteilt werden.
- Subsystem: Ein oder mehrere Fördersysteme, die unter einer gemeinsamen Verwaltung stehen und damit auf derselben Steuerungslogik basieren. Die im Projekt entwickelte kollaborative Plattform strebt eine effiziente Zusammenarbeit zwischen den

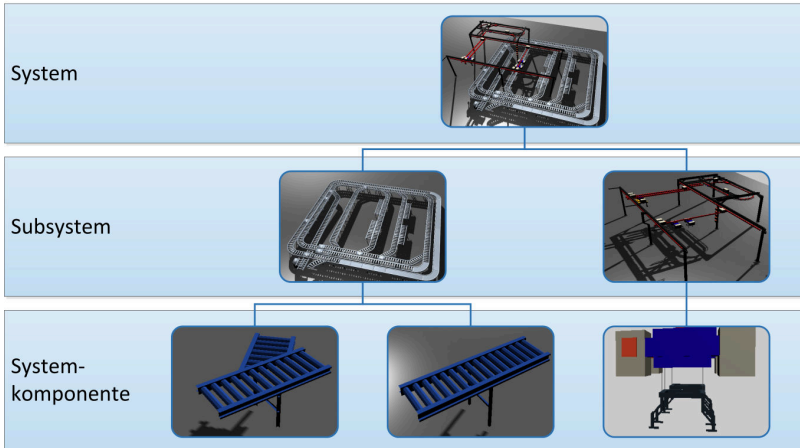


Abbildung 4.3: Hierarchische Struktur eines Materialflusssystems

Subsystemen der verteilten Materialflussteuerung an. Ein Subsystem kann beispielsweise eine Elektrohängebahn (EHB) mit autonomen Fahrzeugen sein (siehe Versuchsanlage fml in Kapitel 9).

- **Systemkomponenten:** Einzelne, autonom agierende Fördertechnikelemente wie Rollenbahnen, Elektrohängebahnkatzen oder Regalbediengeräte. Bei dezentral gesteuerten Anlagen sind Softwareagenten für die Steuerung dieser Systemkomponenten zuständig.

Die Agentensteuerung, die auf der dritten Ebene angesiedelt werden kann, übernimmt Aufgaben wie Auftragsdisposition, Routenplanung oder Ressourcenallokation. Diese komplexen Steuerungsaufgaben werden durch Softwareagenten realisiert, welche sich als autonome und untereinander kommunizierende Softwarekomponenten, die kooperieren, um spezifische Ziele zu erreichen, beschreiben lassen. Die einzelnen Module tauschen folglich untereinander Informationen aus, um Entscheidungen, z. B. im Rahmen der Routenplanung, treffen zu können. Für die Umsetzung der Agentenlogik werden Multiagentensysteme verwendet. Diese Multiagentensysteme erfordern eine Laufzeitum-

gebung – die so genannte Agentenplattform –, die in der Regel auf einem PC-ähnlichen Rechner umgesetzt werden kann. Die Kommunikation zwischen den Ebenen der Maschinen- und der Agentensteuerung wird durch eine Abstraktionsschicht, oder Middleware realisiert. Diese übersetzt maschinenspezifische Daten in eine Sprache, welche die Softwareagenten verwenden können.

4.2 Fazit

Der hier vorgestellte kollaborative Engineeringprozess sichert mit seinen Tätigkeiten die Vorteile, die bei der Entwicklung dezentraler Materialflusssysteme entstehen. Insbesondere die projektunabhängigen Vorarbeiten können gut in eine Kollaborationsplattform eingebracht werden. So kann in den Projekten ein Großteil der Arbeiten frühzeitig und Inhouse geschehen. Die Annahme einer gemeinsamen Datenplattform und die Möglichkeit der synchronen Bearbeitung in den einzelnen Projektphasen wird erst durch eine geeignete Kollaborationsplattform ermöglicht.

5 Anforderungen an die Kollaborationsplattform

In diesem Kapitel werden das Problemfeld der computergestützten Planung von Intralogistiksystemen strukturiert und eine Bestimmung der Anforderungen an ein softwarebasiertes Kollaborationssystem vorgenommen. Die Spezifikation dieser Anforderungen sichert die technische Eignung der KoDeMat-Plattform für den Planungseinsatz und bildet die Basis für die Softwarearchitektur (vgl. Kapitel 6). Um ein Verständnis der Anwendung der Plattform und der allgemeinen Interaktion zwischen den Akteuren des Planungsprozesses (siehe Abschnitt 5.1.1) zu ermöglichen, wird zusätzlich eine Analyse der logistischen Prozesse und Funktionen durchgeführt.

Damit gewährleistet ist, dass praxisnahe Methoden in die technische Lösung der KoDeMat-Plattform integriert werden, fokussiert sich die Analyse auf potenzielle Anwendungsfälle und die Entwicklung eines Referenzszenarios (siehe Abschnitt 5.1.2). Das Referenzszenario deckt einen repräsentativen, praxisnahen Anwendungsfall ab. Diese Vorgehensweise ist in Abbildung 5.1 illustriert. Eine ähnliche Vorgehensweise wurde bei der Gestaltung von kollaborativen Software-Plattformen für logistische Transportsysteme angewendet (vgl. [Tjo11]).

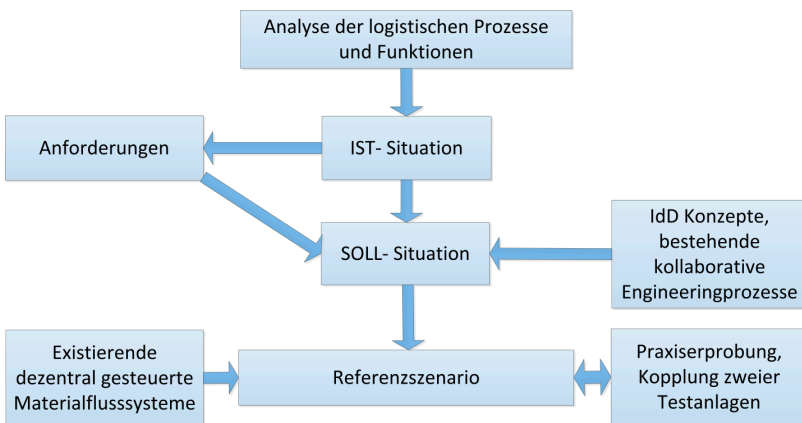


Abbildung 5.1: Vorgehensweise für die Gestaltung der KoDeMat-Plattform

5.1 Anwendungsfall

Im BMBF-Projekt *Internet der Dinge* (vgl. [GHM10]) wurde der Planungs- und Inbetriebnahmeprozess intralogistischer Anlagen untersucht. Dabei wurde das Problemfeld einer unternehmensübergreifenden Planung und Realisierung von dezentral gesteuerten Materialflusssystemen, insbesondere bei heterogenen Großprojekten, identifiziert. Dabei stellt sich die Frage, wie die technischen Vorteile einer dezentral gesteuerten Anlage auch in verteilten Projekten realisiert werden können. In einem solchen Projekt (vgl. Abbildung 5.3) arbeiten mehrere Hersteller dezentral gesteuerter Subsysteme oder Systemkomponenten (siehe Abschnitt 4.1) sowie Systemintegratoren zusammen und erstellen ein Gesamtsystem.

Im Unterschied zu einem Projekt mit homogener Softwarearchitektur, werden hier in den einzelnen Subsystemen unterschiedliche Ansätze einer dezentralen Steuerung ermöglicht. Es kann daher nicht mehr von standardisierten Schnittstellen ausgegangen werden. Vielmehr müssen diese einzeln spezifiziert, abgestimmt und implementiert werden. Bei einem heterogenen Großprojekt besteht die Möglichkeit, dass die Vorteile der dezentralen Steuerungssysteme, die sich üblicherweise aus der Verlagerung von projektbezogenen Arbeitsschritten in das Projektvorfeld ergeben, stark abgeschwächt werden. Einerseits besteht die Gefahr, dass durch unzureichende Transparenz die Zusammenarbeit der beteiligten Unternehmen behindert wird. Zum anderen kann sich der Grad der Wandlungsfähigkeit der eingesetzten Subsysteme bei der Integration in das Gesamtsystem reduzieren.

5.1.1 Akteure

Bei der Planung von Materialflusssystemen arbeiten eine Vielzahl an verschiedenen Unternehmen und Personen zusammen. Dabei können drei verschiedene Hauptrollen (siehe Abbildung 5.2) identifiziert werden, die im folgenden als Akteure bezeichnet werden:

- **Betreiber:** In der Regel der Auftraggeber des Planungsprojekts und der zukünftige Besitzer und Betreiber des Materialflusssystems.
- **Hersteller:** Entwickelt eine oder mehrere fördertechische Systeme sowie deren dezentrale oder zentrale Steuerung.

- Systemintegrator: Unternehmen, die sowohl Planungsschritte ausführen, als auch Software- und Hardwareanpassungen an den Materialflusssystemen durchführen und in die bestehende Anlage integrieren.

Bei der Planung vertreten die Akteure oft unterschiedliche Positionen und Interessen. Im Rahmen der kollaborativen Planung können die Grundfunktionen des Systems von allen Akteuren gleichermaßen verwendet und zur Abstimmung genutzt werden.

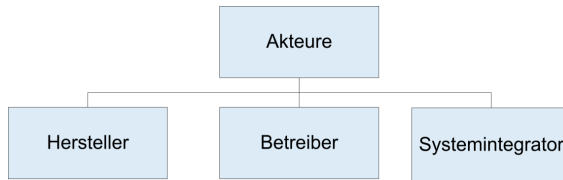


Abbildung 5.2: Akteure, die bei einem intralogistischen Projekt zusammenarbeiten

5.1.2 Referenzszenario

Ein mögliches repräsentatives Referenzszenario beschreibt ein Neu- oder Umplanungsprojekt eines Intralogistiksystems, welches von mehreren Unternehmen durchgeführt wird, die nicht-kompatible dezentrale Steuerungen einsetzen. Im Rahmen des Planungsprojekts sollen die Subsysteme im Vorfeld soweit zusammengeschaltet und als Gesamtsystem getestet werden, dass die Inbetriebnahmephase beim Kunden weitgehend die vorteilhafte zeitliche Verkürzung beibehält, die im Fall eines homogenen Systems zu erwarten wäre. Abbildung 5.3 illustriert ein solches beispielhaftes Referenzszenario. In diesem Planungsprojekt arbeiten verschiedene Hersteller bei der Integration eines neuen Fördertechniksystems in ein beim Kunden existierendes Materialflusssystem zusammen.

Es kann von einer großen räumlichen Entfernung der Standorte der beteiligten Unternehmen ausgegangen werden, die einer dauerhaften, direkten Zusammenarbeit entgegensteht. Die Projektpartner können jedoch über ein Virtual Private Network (VPN) verbunden sein und ihre Kommunikation gesichert und verschlüsselt abwickeln. Durch eine

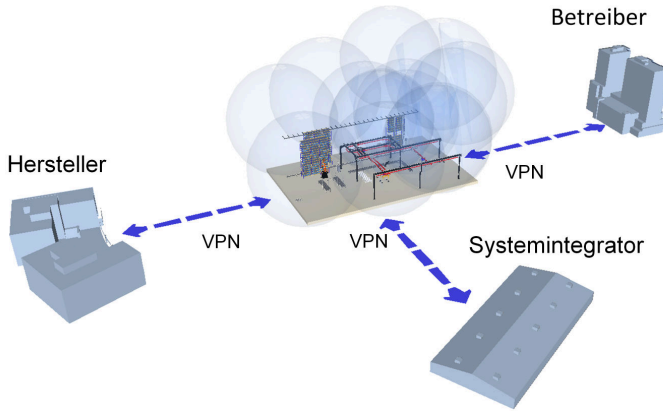


Abbildung 5.3: Referenzszenario der KoDeMat-Plattform für ein heterogenes Großprojekt

kollaborative Planungsplattform werden die Partner gemeinsam eine Simulation und Visualisierung des Gesamtsystems erstellen und zur Planung nutzen. Diese Visualisierung soll insbesondere die separaten Anzeigen der Simulationsumgebungen zusammenfassen und damit der visuellen Validierung der Funktionalität des Gesamtsystems dienen.

Neben der Visualisierung steht die gemeinsame Entwicklung von Schnittstellenspezifikationen im Fokus des Szenarios. Im Unterschied zu einer klassischen, rein textuellen Beschreibung soll die Spezifikation auf maschinenverarbeitbaren Datenmodellen basieren. Eine solche Spezifikation kann als Ausgangspunkt für die Generierung von Programmen dienen, welche genau dieser Spezifikation entsprechen. Somit entfallen Zeitverzögerungen, die durch fehlerhafte, manuelle Implementierungen von rein textuellen Spezifikationen entstehen. Im Rahmen des Szenarios sollen Telegrammdefinitionen, Protokoll-Abläufe und Zustandsautomaten abgebildet werden, die an den Übergabepunkten zwischen den Subsystemen verwendet werden, um einen reibungslosen Materialfluss zu gewährleisten.

Die angestrebte Kollaborationssoftware soll in dem Referenzszenario ab der Grobplanungsphase eingesetzt werden können, also bereits zu

einem Zeitpunkt, an dem die unterschiedlichen Subsysteme noch nicht zusammenarbeiten können. Die Nutzung findet bis hin zur Inbetriebnahme statt. Ein Export der erarbeiteten Daten (wie z.B. den Schnittstellenspezifikationen) kann als Dokumentation des Planungsprozesses dienen und auch für Fortschrittspräsentationen gegenüber dem Kunden eingesetzt werden.

Die Softwareanforderungen können in drei Kategorien unterteilt werden: Geschäftsanforderungen, Benutzeranforderungen und funktionale Anforderungen. Weiterhin können jedem System einige nicht-funktionale Anforderungen zugeordnet werden. Dabei bauen die funktionalen Anforderungen aufeinander auf, während die nicht-funktionalen Anforderungen an vielen unterschiedlichen Stellen einfließen.

5.2 Geschäfts- und Benutzeranforderungen

Die Geschäfts- und Benutzeranforderungen spiegeln das logistische Einsatzumfeld sowie die Ziele der am Planungsprozess beteiligten Akteure wider. Bei der Entstehung einer Materialflussanlage fallen vielfältige Tätigkeiten an, welche unterschiedliche Anforderungen an die Technik und den Menschen stellen. Kernelement der Analyse der Geschäftsanforderungen sind die Tätigkeiten, die von der Planung bis zum Betrieb und der Erweiterung einer Materialflussanlage erforderlich sind. Aus diesen werden entsprechende Einsatz- und Nutzungsmöglichkeiten für eine Kollaborationssoftware abgeleitet.

Wie bereits beschrieben findet die aktuelle Bearbeitung gemeinschaftlicher Intralogistikprojekte mit einer Vielzahl von Medien statt und ist nicht zuletzt geprägt von einer großen Anzahl an E-Mails, die zwischen verschiedenen Akteuren verschickt werden. Informationen zur Entwicklung der Materialflussteuerung finden sich in Textverarbeitungsdateien, Programmdateien und speziellen Simulationsprojekten. Daher kann der fehlenden Transparenz nur mit einem erhöhten organisatorischen Aufwand begegnet werden, der zu den hohen Kosten der Planung beiträgt. Dieser Effekt verstärkt sich erfahrungsgemäß mit einer höheren Anzahl an Projektbeteiligten, einer größeren räumlichen Distanz der Projektgruppen und einem zeitlichen Versatz in der Durchführung der Arbeitsschritte.

Die grundlegenden Geschäfts- und Benutzeranforderungen können demnach folgendermaßen beschrieben werden:

- Um eine erfolgreiche Integration von kollaborativen Engineeringprozessen in die Praxis sicherzustellen, ist es zielführend, sich an konventionellen, bewährten Prozessen anzulehnen und diese zu unterstützen. Daher ist eine der Voraussetzungen für ein Kollaborationssystem die nahtlose Integration in existierende IT-Strukturen, insbesondere in die relevanten Bereiche der Kommunikation, Koordination und Kooperation.
- Neben der Ausrichtung des Systems auf dezentrale Steuerungsarchitekturen sind auch die bisher dominierenden zentralen Strukturen zu berücksichtigen.
- Allgemeine Kollaborationsinstrumente, wie etwa ein Videokonferenzsystem, eine verteilte Textverarbeitung und ein System zum Austausch von Dateien, sollen zur Verfügung stehen. Dabei ist es zielführend, bestehende und zukünftige allgemeine Kollaborationssysteme, wie z.B. für die Textverarbeitung zu nutzen. Durch die Integration der frei verfügbaren Systeme in die KoDeMat-Plattform kann von den ausgereiften Systemen profitiert werden.
- Ein System zur Versionskontrolle von Text- und Binärdaten soll Anwendung finden und die Bearbeitung der Planungsprozesse vereinfachen und transparenter machen.
- Es soll ein virtueller Test des Gesamtsystems, ohne den physischen Aufbau der Fördertechnik mit der Plattform, ermöglicht werden.

Aus diesen beschriebenen Geschäfts- sowie Benutzeranforderungen können die allgemeinen funktionalen Anforderungen an die Kollaborationsplattform und im Speziellen an die Visualisierung und die Schnittstellenverwaltung abgeleitet werden.

5.3 Funktionale Anforderungen an die Visualisierung

Aufgabe der Visualisierung der kollaborativen Planungsplattform ist eine einheitliche Darstellung der zusammengeführten Daten und Aktionen der einzelnen Akteure. Insofern ist die Visualisierung kein Ersatz für Materialflusssimulationen oder Konstruktionsanwendungen.

Vielmehr sollen die Subsysteme und Systemkomponenten visuell miteinander verknüpft werden, damit die Gesamtanlage betrachtet und bewertet werden kann. Um die Aktionen der Akteure im Kollaborationsprozess zu veranschaulichen, müssen visuelle Metaphern gefunden werden. Aus diesen Überlegungen heraus lassen sich folgende funktionale Anforderungen an die Visualisierung formulieren:

- Es ist notwendig, eine Importmöglichkeit für visuelle Modelle (z. B. 3D-Modelle) und Abbildungen (z. B. Grundriss) zu schaffen, die es erlaubt, existierende Daten in die Kollaborationsplattform einzubinden.
- Mit einfachen Editorfunktionen, wie dem Laden, Verschieben, Drehen und Löschen von 3D-Modellen und Bildern, soll das Zusammenführen unterstützt werden.
- Die Anwesenheit von Akteuren soll mithilfe visueller Metaphern dargestellt werden. Neben den Subsystemen und Systemkomponenten sollen die Akteure Markierungen und Kommentare in die visuelle Umgebung einbringen können.
- Um die Integration heterogener Materialflusssysteme effizient zu ermöglichen, sollen die Akteure die Möglichkeit haben, Schnittstellen zwischen Subsystemen (wie z. B. Übergabepunkte) zu definieren, zu visualisieren und zu bearbeiten.
- Eine standardisierte API-Schnittstelle soll es ermöglichen, eine einfache Anbindung an real existierende Materialflusssysteme und deren Steuerungen zu schaffen. Damit kann beispielsweise die Funktion der aktuellen Anlage und die zukünftige Funktion nach einer Erweiterung visualisiert und getestet werden.

Diese Funktionen der Visualisierung ermöglichen bereits frühzeitig ein 3D-Abbild des Gesamtsystems bereitzustellen. Die Importmöglichkeit für gängige Dateiformate kann dazu genutzt werden, im Projektverlauf immer detailliertere Modelle zu laden. Beispielsweise würden während der frühen Planungsphase nur Platzhaltermodelle eingesetzt, während nach der Auftragsvergabe für die Gewerke die tatsächlichen Konstruktionsmodelle eingesetzt werden können. Die Möglichkeit der Erzeugung von Materialflusstopologien durch die Definition von Übergabepunkten und Objektflussgraphen ist ein Merkmal,

das speziell für diesen Anwendungsfall notwendig ist und in klassischen 3D-Modellierungswerkzeugen fehlt. Die Darstellung der Übergabepunkte ist gleichzeitig die visuelle Metapher für die Schnittstellenverwaltung.

5.4 Funktionale Anforderungen an die Schnittstellen- und Versionsverwaltung

Im Rahmen des kollaborativen Planungsprozesses nehmen die Schnittstellen zwischen den Subsystemen und den Systemkomponenten verschiedener Hersteller eine wichtige Rolle ein. Jedes Subsystem und jede Systemkomponente kann dabei eine oder mehrere Schnittstellen zu anderen besitzen. Für die zielführende Definition und Spezifikation werden die folgenden Anforderungen aufgenommen:

- Schnittstellen sollen definiert und im Materialflusssystem verortet werden können, d. h. es wird ihnen eine Position im Materialflusssystem zugeordnet.
- Die Spezifikation einer Schnittstelle soll aus Telegrammdefinitionen und Protokollfestlegungen bestehen.
- Die Telegramme sollen sich aus einer Sequenz von Zeichen mit einer festen Länge zusammensetzen. Subsequenzen sollen zu einzelnen Datenfeldern gruppiert werden können.
- Die Schnittstellenspezifikationen sollen wiederverwendbar sein. Daher soll die Möglichkeit bestehen, Templates zu definieren, welche eine automatische Generierung von Quellcode für die agentenbasierte Steuerung ermöglichen.

Die obengenannten Anforderungen an die Schnittstellen- und Versionsverwaltung sind in Zusammenarbeit mit den Projektpartnern aus der Industrie erarbeitet worden. Die Schnittstellenverwaltung ermöglicht mit diesen Funktionen einen weitestgehend automatisierten Arbeitsprozess bei der Softwareentwicklung. Insbesondere die automatische Codegenerierung ermöglicht es, Änderungen der Spezifikation direkt in lauffähigen Code zu übertragen.

5.5 Nicht-funktionale Anforderungen

Im Gegensatz zu den funktionalen Anforderungen, die beschreiben was eine kollaborativen Planungsplattform tun soll, legen die nicht-funktionalen Anforderungen fest, welche technischen Eigenschaften notwendig sind, um einen reibungslosen Betrieb zu ermöglichen. Die folgenden aus Expertengesprächen gewonnenen nicht-funktionalen Anforderungen können spezifiziert werden:

- Da es sich in diesem Szenario um ein vernetztes System handelt, bei dem die Akteure der Kollaborationsplattform über große Entfernungen miteinander arbeiten sollen, spielt das gesamte Zeitverhalten des Systems eine wichtige Rolle. Das System muss die synchrone Zusammenarbeit zwischen Akteuren ermöglichen. Dies bedeutet, dass jegliche Änderungen eines Akteurs allen anderen Akteuren zeitnah übermittelt werden müssen. Der Zeitfaktor beeinflusst hier direkt die Fähigkeit des Systems, den notwendigen gleichzeitigen Informationsstand herzustellen.
- Das System muss die asynchrone Zusammenarbeit ermöglichen, bei der die Akteure mit zeitlichem Versatz an den gleichen Datenmodellen Veränderungen durchführen.
- Das System muss portierbar und plattformunabhängig sein. Das bedeutet, dass es ohne hohe Rechnerleistung und auf möglichst mehreren Betriebssystemen laufen können muss.
- Um neue kollaborative Funktionen zu unterstützen, muss eine einfache Erweiterbarkeit des Systems gewährleistet sein.
- Die Nutzung des Systems soll möglichst intuitiv sein. Damit sind Akteure in der Lage, ohne umfassende Schulungen die Kollaborationsplattform zu verwenden.

Die Kollaborationsplattform muss die zeitgleiche oder zeitversetzte Zusammenarbeit mehrerer Akteure unterstützen. Wenn ein Akteur das System verlässt, sollen die Arbeiten von den restlichen Akteuren fortgesetzt werden können. Akteure sollen ebenfalls nachträglich in ein laufendes System eintreten können.

Gleichfalls sollen alle Aktionen, die Akteure durchführen, gespeichert werden. Die Historie aller Änderungsvorgänge soll angezeigt, kommentiert sowie rückgängig gemacht werden können.

Die Zielsetzung ist, eine Nachvollziehbarkeit für alle beteiligten Akteure herzustellen, die es ermöglicht, einen fachlichen Konsens herzustellen. Weiterhin soll die Möglichkeit bestehen, Daten wie die Anlagentopologie und Schnittstellenspezifikationen in ein plattformunabhängiges Format (z. B. XML) zu exportieren und zu importieren. Im Folgenden werden weitere spezifische funktionale Anforderungen an die Visualisierung und Schnittstellenverwaltung beschrieben.

Obwohl die Anforderungen der einzelnen Teilbereiche der Kollaborationsplattform bereits in Teilen von existierenden Werkzeugen abgedeckt werden (z. B. Multi-User-Texteditoren), bildet das nahtlose Ineinandergreifen der Einzelsysteme erst die Grundlage für eine reibungslose Arbeit mit dem System.

6 Spezifikation und Implementierung einer kollaborativen Softwareplattform

Aufgrund der fehlenden Standardisierung und Heterogenität der Steuerungsansätze im Bereich dezentraler intralogistischer Materialflusssysteme ist die Integration und Interoperabilität von mehreren Subsystemen eine komplexe Aufgabe, die zudem eine große Menge an Ressourcen erfordert. Um die dabei entstehenden Herausforderungen zu bewältigen, bedarf es kollaborativer Softwarewerkzeuge, die die Planungsprozesse sowie die Implementierung und den Betrieb von heterogenen intralogistischen Systemen unterstützen. Werkzeuge zur Kollaboration dienen in erster Linie dazu, Missverständnisse zwischen einzelnen Akteuren zu verhindern. Sie ermöglichen zudem ein effektives, gemeinsames Arbeiten an einem Projekt. Im Gegensatz zu einem Werkzeug für einzelne Benutzer verfügen kollaborative Werkzeuge über Mechanismen, die eine sinnvolle, systematische Zusammenarbeit erlauben. Dazu gehören die Nachvollziehbarkeit von Änderungen, eine eindeutige Identifikation der Teilnehmer sowie die Möglichkeit Kommentare zu hinterlassen.

Damit die KoDeMat-Plattform diese Funktionen anbieten kann, wurde eine Softwarearchitektur für die Verteilung von Daten entwickelt. Die Softwarearchitektur bildet damit die Basis für die Schnittstellenverwaltung (siehe Kapitel 8) und die 3D-Visualisierung (siehe Kapitel 7). Im folgenden Abschnitt wird die Softwarearchitektur der Plattform beschrieben.

6.1 Softwarearchitektur

Die KoDeMat-Plattform ist modular aufgebaut und basiert auf einer serviceorientierten Schichtenarchitektur (siehe [Erl09]). Sie besteht aus den folgenden Schichten (siehe Abbildung 6.1):

- Die *Front-End-Schicht* bietet einen Zugangspunkt mit Anpassungsmöglichkeiten für die einzelnen Benutzer, einschließlich integrierter Einrichtungen für den synchronen Informationsaustausch zwischen unterschiedlichen Partnern. Die Software wird

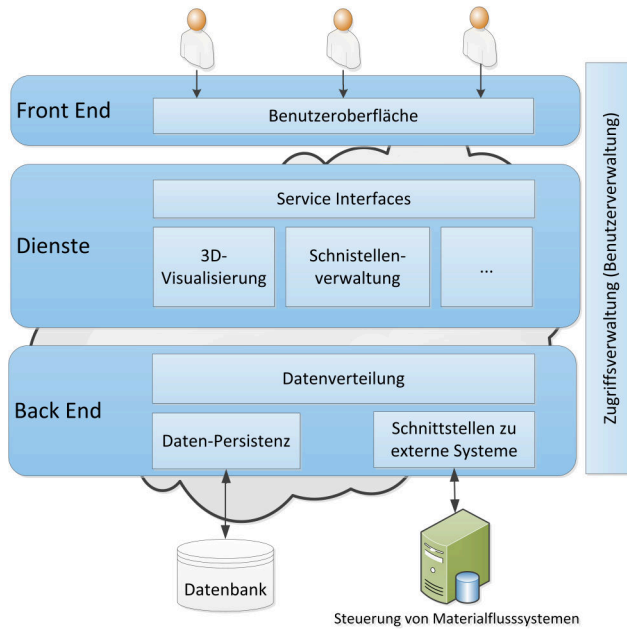


Abbildung 6.1: Softwarearchitektur der KoDeMat-Plattform

als selbstständige Desktopanwendung gestaltet, eine Erweiterung zu einer browserbasierten Lösung ist aufgrund der serviceorientierten Architektur möglich.

- Innerhalb der Schicht *Dienste* (engl. Service-Layer) befinden sich Softwarekomponenten, die die Funktionen der KoDeMat-Plattform erfassen. Diese werden als *Module* bezeichnet. Die beiden Hauptmodule, die implementiert wurden, sind das Planungswerkzeug mit 3D-Visualisierung und das Schnittstellenverwaltungsmodul.
- Die Back-End-Schicht ist für die Synchronisierung der Daten zwischen den Clients zuständig und basiert auf einem verteilten Cluster. Das Back-End unterstützt die Integration externer Systeme, z. B. agentenbasierter Materialflusssysteme oder Datenbanksysteme. Zudem bietet es die Möglichkeit den aktuellen

Zustand einer Sitzung zu exportieren und an einem späteren Zeitpunkt wiederherzustellen. Als Sitzung ist die synchrone Zusammenarbeit von mehreren Akteuren in der Kollaborationsplattform bezeichnet.

Bei der Entwicklung der Softwareplattform werden modulare Prinzipien angewendet. So sind einzelne Softwarekomponenten, wie zum Beispiel das Softwarewerkzeug mit 3D-Visualisierung und die Schnittstellenverwaltung, getrennt einsetzbar. Ziel ist es, dass einerseits eine Wiederverwendbarkeit von Programmteilen nach deren erstmaliger Implementierung gewährleistet wird, aber auch Teilergebnisse in die IT-Infrastruktur potenzieller Nutzer aus der Industrie integriert werden können. Die gesamte Software ist auf frei verfügbare Open-Source-Softwarebibliotheken gesetzt und ist in der Programmiersprache Java geschrieben.

6.2 Verteilung der Daten

Besonders wichtig für die kollaborative Bearbeitung von Daten mit der KoDeMat-Plattform ist deren Verteilung. Die Plattform ermöglicht zwei Formen der Zusammenarbeit zwischen Akteuren: die synchrone sowie die asynchrone Zusammenarbeit. Bei der synchronen Zusammenarbeit bearbeiten alle Akteure gleichzeitig die Daten. Bei der asynchronen Zusammenarbeit findet die Bearbeitung der Daten zu unterschiedlichen Zeitpunkten statt.

Um die beiden Formen der Zusammenarbeit zu realisieren, setzt sich die entwickelte Softwarearchitektur der KoDeMat-Plattform aus zwei Bestandteilen zusammen: dem synchronen Systemkern und dem asynchronen Systemkern. Ihr Unterscheidungsmerkmal ist die zeitliche Synchronität der Abläufe sowie die jeweilige Dauer der Nutzung. Der asynchrone Systemkern ist als klassischer Server langfristig verfügbar und hält alle Projektstände als Versionen bereit. Am Anfang einer Sitzung wird der synchrone Systemkern gestartet. Er fordert daraufhin Daten vom asynchronen Systemkern an. Benutzer können während der Sitzung Daten kollaborativ bearbeiten. Wird eine Sitzung beendet, werden alle im Rahmen der Sitzung veränderten Daten des synchronen Systemkerns wieder auf dem asynchronen Systemkern gespeichert. Für die Verwaltung der Daten wurden für jeden Systemkern entsprechende Datenmodelle entworfen, diese werden in diesem

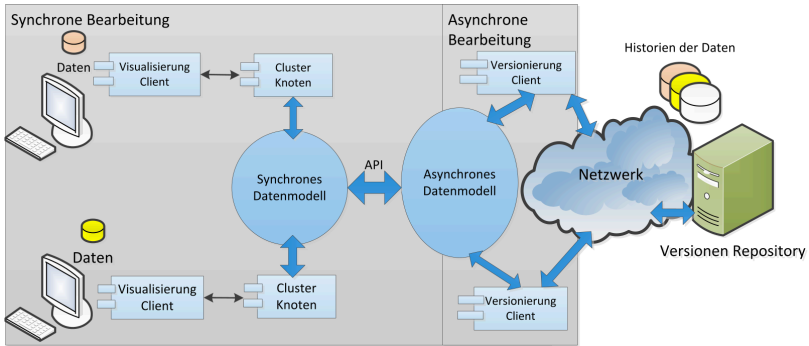


Abbildung 6.2: Konzept der Datenverteilung und Versionsverwaltung der KoDeMat-Plattform

Abschnitt beschrieben. Das oben genannte Konzept ist in Abbildung 6.2 dargestellt.

6.2.1 Synchronisierung der Onlinedaten

Für die rechtzeitige Verteilung der relevanten Daten, die eine synchrone Bearbeitung von kollaborativen Aufgaben erlauben, ist eine geeignete Datenverteilungssoftware erforderlich. Zu diesem Zweck wird ein *In-Memory-Datenbanksystem* genutzt. Das System setzt eine Clusterarchitektur ein, bei der die Daten auf verschiedene Rechner verteilt werden. Im Gegensatz zu traditionellen Datenmanagementsystemen, für die ein oder mehrere Festplattenlaufwerke als primärer Speichermechanismus dienen, nutzen *In-Memory-Datenbanksysteme* den Arbeitsspeicher als Datenspeicher (siehe [LNF13], [LRR⁺13]). *In-Memory-Datenbanken* sind wesentlich schneller und ihre Zugriffszeiten sind besser vorhersagbar als die auf Festplatten zugreifenden Datenbankmanagementsysteme.

Allerdings ist die Speicherkapazität der Arbeitsspeicher verglichen mit der Speicherkapazität der Festplatte kleiner. Durch das so genannte *Grid-Computing* bzw. durch ein *In-Memory-Data-Grid-System* kann die Speicherkapazität erhöht werden. *Data-Grids* basieren auf Clusterarchitekturen, mittels derer die Daten zwischen mehreren Knoten verteilt und redundant gespeichert werden. Dies bedeutet, dass bei

dem Ausfall eines Rechnerknotens die Daten nicht verloren gehen. Eine solche aus Rechenzentren bekannte Architektur ermöglicht es die Vorteile einer zentralen Datenhaltung, wie sie im klassischen Client-Server-System anzutreffen ist, mit der Flexibilität von verteilten Systemen zu verbinden.

Name	Hersteller	Plattform	Open-Source
Gemfire	VMware	Java	Nein
GridGain	DataGrid	Java, .NET, C++	Nein
Oracle Coherence	Oracle	Java	Nein
Hazelcast	Hazelcast	Java, C++	Ja
XAP	Gigaspace	.NET, C++	Nein
Ehcache	Terracotta	Java	Ja
Infinispan	JBoss	Java	Ja

Tabelle 1: In-memory Datenbanksysteme Lösungen (vgl. [Pri13])

Im Bereich In-Memory-Datenbanksysteme gibt es verschiedene Lösungen, die sich im Markt etabliert haben (siehe Tabelle 1). Die Mehrheit der Systeme wird für *Datenbank-Caching* [ALK⁺02] eingesetzt. Für diese Arbeit wird die Open-Source-Plattform Hazelcast [Vee14] ausgewählt. Hazelcast ist nicht so stark auf den Web-Service-Bereich ausgerichtet, sondern wird häufig als ein verteiltes Speichersystem eingesetzt. Mittels Hazelcast wird ein Cluster aus vernetzten Java-Softwarekomponenten aufgebaut (siehe Abbildung 6.3). Jedes Mitglied dieses Clusters wird als *Netzwerk-Knoten* bezeichnet. Die Netzwerk-Knoten werden für die Verteilung der Daten zwischen den Instanzen (Clients) der KoDeMat-Plattform eingesetzt. Hazelcast unterstützt die folgenden Funktionen im Bereich der Datenverteilung:

1. Dynamische Skalierbarkeit: Aufteilung (auch Partition genannt) von Datenstrukturen wie Java-Hashmaps in sortierten und unsortierten Listen. Die Partitionen sind gleichmäßig über alle Knoten verteilt. Jeder Knoten trägt einen Teil der gesamten Daten im Cluster sowie Backupdaten anderer Netzwerkknoten. Wenn ein neuer Knoten dem Cluster beitrifft, bekommt er einen Teil der gesamten Daten im Cluster.

2. Hohe Verfügbarkeit: Das Backup eines Netzwerkknotens wird automatisch erstellt und ist auf anderen Knoten gespeichert. Beim Ausfall eines Knotens sind dessen Daten somit weiterhin verfügbar. Es gibt keinen Knoten, dessen Ausfall den Ausfall des gesamten Systems nach sich zieht. Ein *Single Point of Failure* existiert somit nicht.
3. Hohe Geschwindigkeit: Das In-Memory-System kann Tausende Arbeitsprozessen pro Sekunde durchführen.

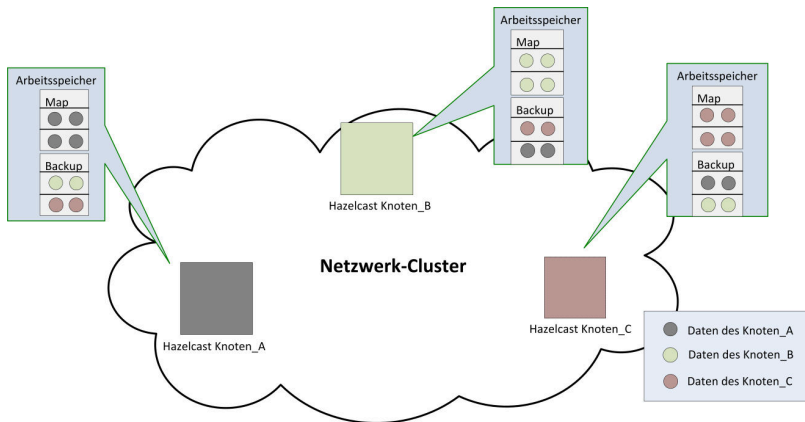


Abbildung 6.3: Cluster des In-Memory-Datenbanksystems von Hazelcast (vgl. [Vee14])

Gestaltung eines Datenmodells für die synchrone Bearbeitung von Daten

Für die Entwicklung einer verteilten Softwareplattform und der Realisierung von kollaborativen Funktionen müssen in erster Linie die zu synchronisierenden Inhalte definiert werden. Basierend auf der Anforderungsanalyse (Kapitel 5) wird ein Konzept eines Datenmodells für die Softwareplattform erstellt. Das Modell erfasst den Anwendungsbereich der intralogistischen Materialflusssystemplanung und kann mittels eines UML-Klassendiagrammes dargestellt werden. Abbildung 6.4 zeigt das Datenmodell der KoDeMat-Plattform. Diese wird im Folgenden beschrieben.

Die Netzwerkknoten des Clusters werden in der Datenmodell als *Knoten* bezeichnet. Jedes Datenobjekt eines Knotens kann seine Daten in das System einfügen und synchronisieren. In der Cluster-Netzwerkarchitektur hat jeder Knoten Zugriff auf die Daten, die ein anderer Knoten im Cluster zur Verfügung stellt. Es können zwei Typen von Knoten im Netzwerk unterschieden werden: *Benutzer* sind menschliche Akteure der Plattform und *System* erfassen die mechatronischen Systemkomponenten einer Anlage (siehe Abschnitt 4.1). Anhand der serviceorientierten Architektur der Plattform sind *Dienste* Funktionen, die im Netzwerk angeboten werden. Die Dienste beziehen sich auf verteilte Daten und können von Knoten ausgeführt werden. Beispiele von Diensten sind die Visualisierung und die Benutzerverwaltung.

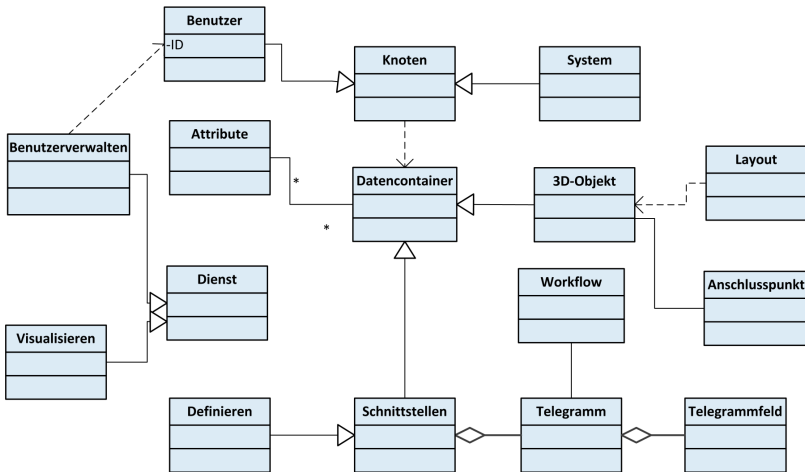


Abbildung 6.4: Datenmodell der KoDeMat-Plattform

Die Ontologie basiert auf dem Konzept eines Datencontainers. Ein Datencontainer ist ein Objekt, das zwischen Knoten synchronisiert wird. Es ist mit einer eindeutigen und globalen ID verknüpft und besitzt ein oder mehrere Attribute. Jeder Datencontainer ist durch ihre Attribute vollständig beschrieben. Die Datencontainer dient zur Aggregation der Attribute. Es wurden zwei Typen von Datencontainer eingeführt, um die Funktionalitäten der Plattform realisieren. Dies ist die 3D-Visuelle-Objekt und der Datencontainer für Schnittstellen. Für die

Funktionen der 3D-Visualisierung wird die 3D-Visuelle-Objekt verwendet. Diese Datenobjekt schließt die Eigenschaften der visuellen Modelle ein und ermöglicht Editorfunktionen, wie das Laden, Verschieben, Drehen und Löschen von 3D-Objekten und Bildern. Ihre Attribute beinhalten Translation, Rotation und Skalierung von Vektoren und Markierungen. Die Attribute der Markierung beschreiben eine Art von Objekt Meta-Information, die eine Erzeugung von Beschriftungen und Kommentaren in der visuellen Umgebung ermöglicht (siehe Abschnitt 7.1).

Neben der 3D-Visualisierung ist ein wesentlicher Teil des Datenmodells die Abbildung der Schnittstellen eines Materialflusssystemes. Zu diesem Zweck werden die Schnittstellen- Objekte verwendet. Bei der Aufgabe der Datenmodellierung von Schnittstellen zwischen heterogenen Materialflusssystemen treten die zwei folgenden Herausforderungen auf. Einerseits muss das Datenmodell spezifisch genug sein, um möglichst viele Anwendungsfälle ohne Duplikate effektiv zu erfassen. Andererseits muss es auch auskömmlich generisch sein, um die Erweiterbarkeit des Modells zu unterstützen und die Steuerungsarchitektur von mehreren dezentralen Systemen effektiv zu beschreiben. Zu einem Datencontainer in dem Modell werden Schnittstellen zu Datenobjekten aus Telegrammen gebildet, diese beinhalten ein oder mehrere Telegrammfelder. Um die komplexen Arbeitsabläufe, die von Agentensystemen ausgeführt werden, modellieren zu können, wird das Workflow-Datenobjekt verwendet. Ein Workflow enthält mehrere Telegramme, die zwischen den Agentensystemen (siehe Abschnitt 8.1) ausgetauscht werden. Wie bereits erwähnt ist ein Vorteil der entwickelten Datenmodellierung ihr generischer Aufbau. Dadurch können verschiedene Systeme und kollaborative Prozesse abgebildet werden. In Bezug auf die Modellierung des Anlagenlayouts ist eine Sammlung visueller Komponenten und deren Beziehungen als ein Layout in dem Datenmodell abgebildet. Anhand des Layout-Objekts ist der Daten-Import/Export Funktion implementiert (siehe Abschnitt 6.4). Durch die oben erwähnte Modellierung kann das Datenmodell des Systems erweitert werden. Dies wird dadurch ermöglicht, dass mehrere Typen von Datencontainer mittels ihrer Attribute abgebildet sind.

6.2.2 Synchronisierung und Versionsverwaltung der Offlinedaten

Die Ergebnisse der synchronen Datenbearbeitung werden während einer Sitzung (z. B. die Anlagentopologie, Schnittstellenspezifikationen) langfristig gespeichert und damit auch einer Version zugeordnet. Der Begriff *Versionsverwaltung* beschreibt Funktionen, die implementiert werden, um die Nachverfolgung und das Zurücksetzen von Änderungen am Datenmodell vorzunehmen. Das Ergebnis ist eine lückenlose Dokumentation des Planungsprozesses sowie die Unterstützung kollaborativen Arbeitens. Zudem können externe Systeme, wie z. B. relationale Datenbanksysteme, auf die Daten zugreifen. In Rahmen der Forschungsarbeiten werden diese beiden Funktionen in die entwickelte Softwareplattform integriert.

Erstellen von Objekthistorien und Zurücksetzen von Änderungen

Eine Herausforderung bei der Erstellung der Softwarearchitektur ist die Modellierung der Versionsverwaltung. Einerseits muss diese bei gleichzeitiger Bearbeitung echtzeitnahe Leistungen erbringen, andererseits muss im gesamten Projektverlauf auch die Verfolgung von Änderungen über größere Zeiträume ermöglicht werden. In der KoDeMat-Plattform wird die Grundlage für die Implementierung von Versionsdaten realisiert. Dazu wird das Framework um das sogenannte *Command Pattern* [Gam04] erweitert, welches jegliche Änderungsaktionen als ein Datenobjekt kapselt. Dieses Objekt enthält Informationen über die ausführenden Akteure und einen Zeitstempel. Die einzelnen Versionsschritte werden pro Datencontainer gespeichert. Das Nachvollziehen der Sequenz dieser Schritte ermöglicht die vollständige Nachverfolgung von Änderungen eines einzelnen Datenobjekts. Ein Versionsschritt kann beispielsweise die Änderung der Position eines 3D-Objektes sein, der zu einem Anlagelayout gehört. Um das Konzept visuell darzustellen, zeigt der Screenshot in der Abbildung 6.5 ein Szenario bei dem zwei Benutzer zusammenarbeiten, um die Position eines Fördersystems zu ändern. Die Änderungen der Attribute (z. B. Position, Ausrichtung) sind in einer Tabelle in Abbildung 6.6 aufgelistet – so kann ein Benutzer einsehen, welche Daten wann und von wem geändert wurden. Dies ermöglicht eine leichtere Diagnose von Fehlerfällen und erlaubt das Nachvollziehen auch komplexer Vorgänge. Für jede synchronisierte Datenobjekt im System wird eine eigene Historie erstellt.

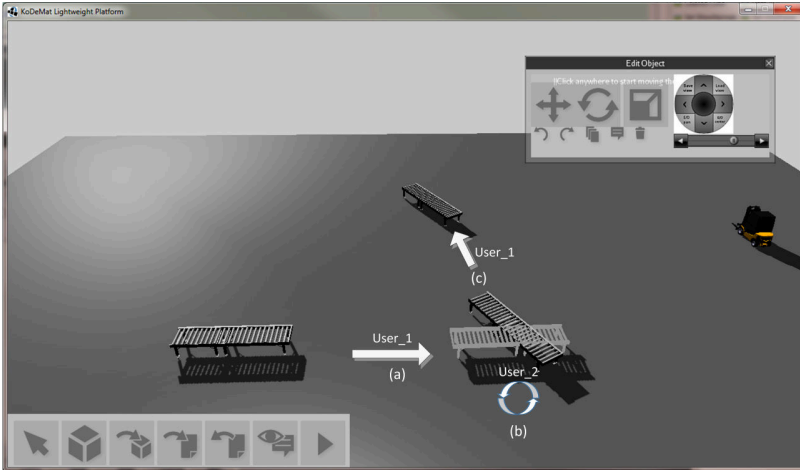


Abbildung 6.5: Beispiel einer synchronen Bearbeitung von 3D-Objekten von zwei Benutzern (User_1 und User_2) (a) User_1 verschiebt das 3D-Objekt eines Fördersystem zu Position B (b) User_2 dreht das Objekt nach rechts. (c) Der User_1 verschiebt das Objekt zu Position C.

tab1	Username	Action
0	User_2_fml	[25/09/2014 16:18:17] - set type of to (Model, /CeMAT_Assets/Rollenfoerederer_pallette/RF_pallette;3o)
1	User_2_fml	[25/09/2014 16:18:17] - set translation to (-49.44806, -0.019721985, -27.23365)
2	User_1_FLW	[25/09/2014 16:19:54] - set translation to (-18.403717, -0.019721985, -21.990623)
3	User_2_fml	[25/09/2014 16:22:26] - set rotation to (0.0, 30.0, 0.0)
4	User_1_FLW	[25/09/2014 16:22:46] - set translation to (9.775893, -0.019729614, -8.214012)

Abbildung 6.6: Änderungshistorien des 3D-Objektes für das Szenario der Abbildung 6.5. Im Visualisierungswerkzeug werden Zeitstempel, Änderungsinformation und Benutzername für jede neue logische Einheit erweitert.

Gestaltung einer ereignisgesteuerten Architektur für die Aufzeichnung von Objekthistorien

Die Funktionen zur Verfolgung von Benutzeraktionen, dem Zurücksetzen von Änderungen sowie eine Aufzeichnung von Objekthistorien erfordern eine flexible und ereignisgesteuerte Softwarearchitektur. Diese wird in diesem Abschnitt vorgestellt. Bei einem *Anfrage-Antwort*-

System (engl. Request / Reply) (siehe Abbildung 6.7 (links)) werden während eines Anfrageprozesses explizit Informationen angefordert. Eine ereignisgesteuerte Architektur bietet hingegen für die Systeme einen Mechanismus der eine dynamische Reaktion bei Eintritt von Ereignissen ermöglicht. In einer solchen Architektur werden Ereignisse (engl. Events) von Softwareprozessen, auch *Event-Erzeuger* genannt, erzeugt. Sobald die Ereignisse von anderen Prozessen eintreten sind, erhalten die *Event-Verwalter* die Ereignisse (siehe Abbildung 6.7 rechts). Da der Event-Erzeuger keine Verantwortung für die Behandlung des Ereignisses übernimmt, ist für diesen die Information, durch welches Objekt das Ereignis abgefertigt wird, nicht notwendig. In gleicher Weise ist für den Event-Verwalter nicht relevant, durch welche logische Einheit das Ereignis erzeugt wurde. Dieser benötigt nur die Information, dass er aufgerufen wurde. Diese Dissoziation zwischen Event-Erzeuger und Event-Verwalter ist ein großer Vorteil der ereignisgesteuerten Architektur.

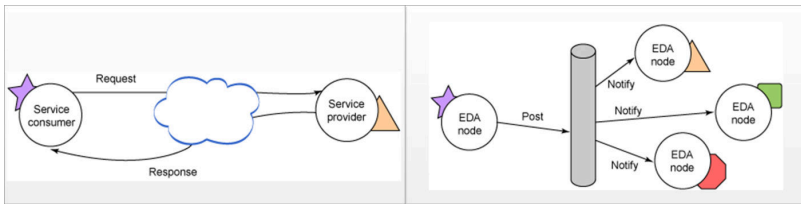


Abbildung 6.7: (l.): serviceorientierte Architektur (engl. SOA), (r.): ereignisgesteuerte Architektur (engl. EDA) [Mar06b]

In der KoDeMat-Plattform werden die genannten Prinzipien einer ereignisgesteuerten Architektur integriert und an die Systembedürfnisse angepasst. Wenn sich die Attribute eines Datencontainer ändern (siehe Abschnitt 6.2.1), werden neue Ereignisse erzeugt. In dem Sinn verhalten sich der Datencontainer als Event-Erzeuger. Die implementierten Dienste (z. B. Visualisierung und Schnittstellenmanagement) stellen hingegen die Event-Verwalter dar. Die Teilnehmer einer Sitzung interagieren mit spezifischen Datencontainer durch komplexe Nachrichten, die so genannten Datenobjekte *Befehle*. Befehle bestehen aus einem Tupel, welches als Informationen die Knoten_ID, der Datencontainer_ID und das Ereignis Objekt beinhaltet. Die Befehle werden zu einer Befehle-Warteschlange (engl. Command-Queue) hinzu-

gefügt, die alle Ereignisse speichert und sortiert. Anschließend werden die entsprechenden Event-Verwalter Objekte über den Auftritt informiert und bearbeiten die Änderungen, die das Event beinhaltet. Die beschriebene Architektur ist in der Abbildung 6.8 illustriert.

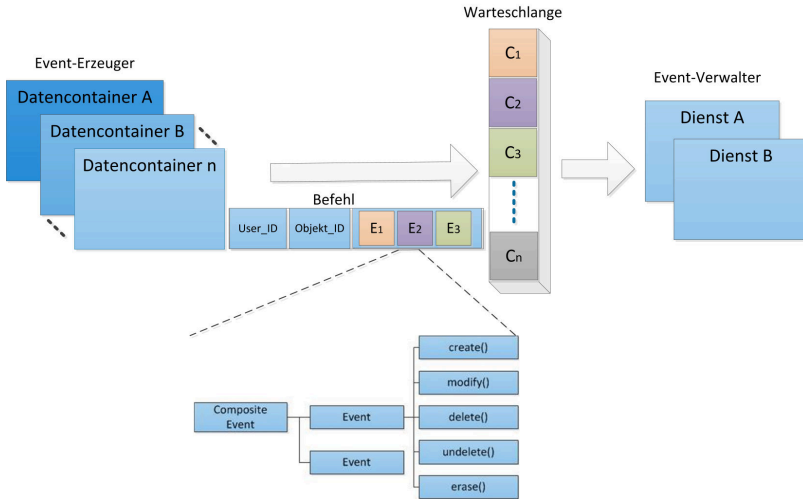


Abbildung 6.8: Ereignisgesteuerte Architektur der KoDeMat-Plattform

Die obengenannten Änderungen beziehen sich auf die Attribute eines Datencontainers. Diese Attribute sind in verteilten Datentabellen (z. B. Java-Hashmaps) der Daten-Verteilungs-Plattform von Hazelcast gespeichert. Änderungsereignisse werden an alle Knoten, die zu einem Netzwerk gehören, geschickt (siehe Abschnitt 6.2.1). Das bedeutet, dass die Größe der Objekte von Bedeutung ist, wenn sich Tabelleneinträge schnell ändern. Diese schnelle Änderung kann zum Beispiel beim Visualisierungsprozess eine Rolle spielen, die häufige Updates von Attributen durchführen. Daher ist es für die Ansprechempfindlichkeit des Systems wichtig, dass die Ereignisse so schnell wie möglich innerhalb des Cluster verteilt und bearbeitet werden. Zu diesem Zweck sind die Strukturen von *Composite-Events* implementiert. Die Composite-Events sind Datenstrukturen die eine Reihenfolge von Ereignissen kapseln. Im Gegensatz zu einer Sequenz einzelner Ereignisse können Composite-Events als *atomisch* charakterisiert werden. Das

heißt, dass alle Ereignisse Änderungen für ein spezifischer Datencontainer verursachen. Ein Beispiel eines solchen ereignisgesteuerten Prozesses ist in Tabelle 2 beschrieben.

Sequenz von Single-Events	Composite-Event (atomisch)
(User_ID, Change_1)	(User_ID, Change_1, Change_2, Change_3)
(User_ID, Change_2)	
(User_ID, Change_3)	

Tabelle 2: Beispiel eines Composite-Event

Lebenszyklus eines synchronisierten Datenobjekts

Der Lebenszyklus eines synchronisierten Datenobjekts beginnt mit seiner Erstellung, die durch einen entsprechenden Befehl vom Client erfolgt. Sobald ein Datencontainer erstellt wird, befindet sie sich in einem *aktivierten* Zustand. Während des aktivierten Zustandes können die Attribute der Datencontainer durch Bearbeitungsereignisse geändert werden. Der Zyklus kann vorläufig mit dem Ereignis *Löschen* beendet werden. Dieses Ereignis versetzt den Datencontainer in einen inaktiven Zustand. Der Datencontainer können jedoch wiederhergestellt werden. Dabei werden alle Attribute auf den Zustand vor der Löschung zurückgestellt. Einer spezifische Eventqueue erlaubt die Wiederherstellung eines Datencontainers zu ihrem ursprünglichen Zustand. Zudem ist es möglich, weitere Änderungen durch zu führen oder der Datencontainer *endgültig* zu löschen. Der komplette Lebenszyklus ist im Zustandsdiagramm in Abbildung 6.9 dargestellt.

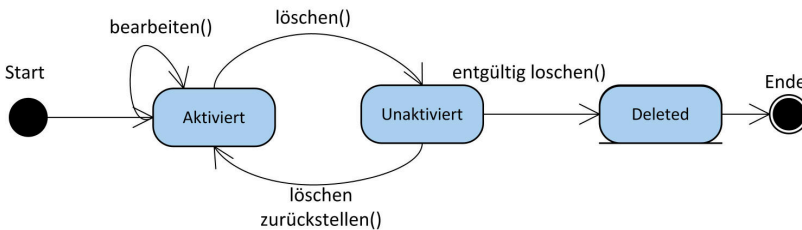


Abbildung 6.9: Lebenszyklus eines Datencontainers

6.3 Ermöglichung einer einheitlichen Visualisierung von 3D-Objekten

Die Funktionen zur Visualisierung sind als ein Dienst implementiert der Ereignisse aus Datencontainer visuell darstellt. Die entwickelte Softwarearchitektur sieht bereits vor, dass für das System relevante Informationen in Datencontainer abgebildet werden (siehe Abschnitt 6.2.1). Diese Daten sind mittels des Clusters zur Datenverteilung (siehe Abschnitt 6.2) an alle Knoten, die dem Cluster zuhören, verteilt worden. Jedes 3D-Objekt ist als ein Datencontainer abgebildet und beinhaltet in Attributen gespeicherte Informationen, die relevant für die 3D-Visualisierung sind (z. B. Position, Ausrichtung). Die binären Dateien der 3D-Objekte sind nicht anhand ihre Größe im Cluster verteilt. Stattdessen verwaltet jeder Knoten dabei lokal die binäre Datei und überträgt nur die Modelldaten mittels der Attribute an die Knoten zur Anzeige. Die Daten des 3D-Objekts werden über die Clusterarchitektur jeweils mit allen beteiligten Knotenrechnern synchronisiert. So werden lokale Änderungen in einem Clientrechner zeitnah an alle weiteren Rechner verschickt. Die 3D-Visualisierung verwendet JMonkey [Kus13], eine Open-Source-Grafikengine, die auf dem OpenGL-Standard aufbaut.

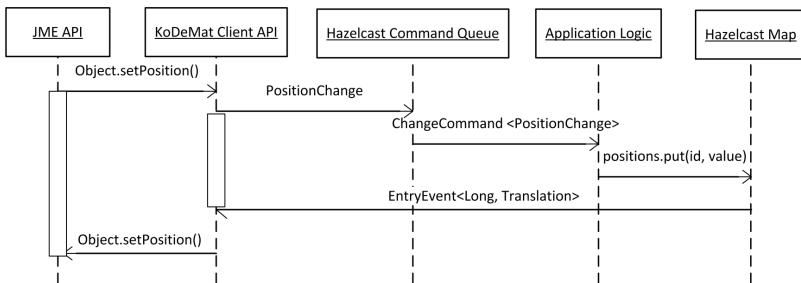


Abbildung 6.10: Kommunikationsdiagramm zwischen Softwarekomponenten bei einer Positionsänderung eines 3D-Objektes

Das in 6.10 dargestellte Kommunikationsdiagramm zeigt den Nachrichtenverkehr zwischen den beteiligten Softwarekomponenten bei einer Verschiebung eines 3D-Objektes in der virtuellen Umgebung. Das Kommunikationsdiagramm im Abbildung 6.10 zeigt den Austausch

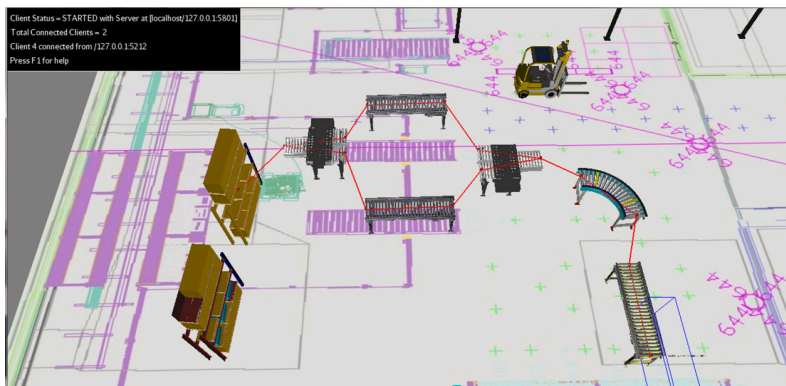
von Informationen zwischen Softwarekomponenten der KoDeMat-Plattform bei einer Änderungsanforderung von ein Objektattribute. Jede Änderung wird an die Command-queue (siehe Abschnitt 6.2.2) angehängt und mit einem entsprechenden Zeitstempel versehen. Die Applikationslogik verarbeitet die Befehle und aktualisiert den Zustand des Datencontainer mittels deren Attribute.

6.4 Importierung / Exportierung von Daten

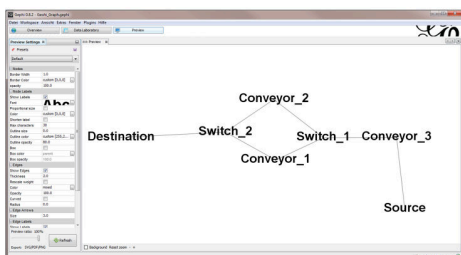
In der Anforderungsanalyse (siehe Abschnitt 5.2) wurde festgestellt, dass die KoDeMat-Plattform die Daten in ein plattformunabhängiges Format exportieren und importieren können muss. Zu diesem Zweck wurde eine Import/Export Funktion implementiert. Durch diese wird es ermöglicht, dass die Daten einer Sitzung in ein XML Format exportiert werden. Extensible Markup Language (XML) ist eine Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten in Form von Textdateien. XML wird u. a. für den plattform- und implementationsunabhängigen Austausch von Daten zwischen Computersystemen eingesetzt [Bra06]. Es gibt viele XML-basierte Formate die heutzutage in verschiedenen Anwendungsbereichen eingesetzt werden. Das Datenmodell der KoDeMat-Plattform bezieht sich auf Beziehungen zwischen Materialflusssysteme, die als hierarchischen Graphen dargestellt werden können. Dafür wird als Daten-Export-Format (Graph Exchange XML Format (GEXF)) (siehe [HB13]) verwendet. GEXF wird eingesetzt, um die Daten und Dynamik von komplexen Netzwerkstrukturen zu beschreiben. Für den Export in einer GEXF Datei wird die hierarchische Struktur eines Gesamtsystems, die im Abschnitt 4.1 präsentiert ist, verwendet.

Beim Einsatz des GEXF als Datenformat besteht die Möglichkeit die exportierte Datei mittels externer Softwarewerkzeuge zu öffnen und zu bearbeiten. Ein solches Werkzeug wurde innerhalb des Gephi Projektes entwickelt und bietet eine interaktive Visualisierungsplattform für Netzwerkstrukturen sowie für dynamische und hierarchische Graphen [BHJ09]. In Abbildung 6.11(b) ist die Visualisierung eines Graphen mit Gephi als Ergebnis einer Anlagetopologie in der 3D-Umgebung des Planungswerkzeugs der KoDeMat-Plattform 6.11(a) illustriert. Weiterhin gibt es die Möglichkeit die Visualisierung eines Graphens in Form einer Datenmatrix auszuführen. Darüber hinaus können die

exportierten Daten in Gephi bearbeitet, gespeichert und erneut auf die KoDeMat-Plattform geladen werden.



(a)



(b)

Nodes	Id	Label	translation
Conveyor_1	7	Conveyor_1	0.059650064, 1.9073486E-6, 8.740E
gurttrennModel	12	gurttrennModel	1.0, 0.7, -0.1
Switch_1	10	Switch_1	-8.417563, 9.536743E-7, 4.689744
weicheModel	16	weicheModel	8.0, 0.7, -0.1
weicheModel2	17	weicheModel2	3.0, 0.7, -0.1
Switch_2	4	Switch_2	0.0, 0.5, 0.0
gurtModel	5	gurtModel	0.0, 0.0, -0.1
Conveyor_2	8	Conveyor_2	-5.533704, 0.0, 5.7406244
gurttrennModel2	14	gurttrennModel2	3.0, 0.7, -0.5
gurttrennModel1	13	gurttrennModel1	3.0, 0.7, -0.1
Conveyor_3	6	Conveyor_3	0.2994721, 2.3941859E-7, 5.623091
kurveModel	11	kurveModel	0.0, 0.6, -0.1
Destination	9	Destination	-9.047445, 0.0, 2.3285675
regalModel	15	regalModel	5.0, 0.7, -0.1
Source	1	Source	

(c)

Abbildung 6.11: (a) Definition einer Anlagentopologie im Planungswerkzeug, (b) Die im GEXF-Format exportierte Topologie kann in einer externen Anwendung wie Gephi dargestellt und bearbeitet werden, (c) Der resultierende Graph kann als Datenmatrix visualisiert werden

Der Export der Daten in ein textuelles Datenformat wie GEXF ermöglicht die Verwendung von externen Versionsverwaltungssystemen für Quelltext. Die Versionsverwaltungssysteme sind im Bereich der Softwareentwicklung sehr verbreitet und dienen zur Erfassung von Änderungen an Dokumenten oder Dateien. Alle Versionen werden in einem *Code-Repository* mit Zeitstempel und Benutzererkennung gesichert und

können somit später wiederhergestellt werden. Hauptfunktionen sind [Bae05]:

- Protokollierungen der Änderungen: Es kann jederzeit nachvollzogen werden, wer wann was geändert hat.
- Wiederherstellung von alten Versionen einzelner Dateien: Somit können versehentliche Änderungen jederzeit wieder rückgängig gemacht werden.
- Archivierung der einzelnen Version eines Projektes: Dadurch ist es jederzeit möglich, auf alle Versionen zuzugreifen.
- Konfliktmanagement: Koordinierung des gemeinsamen Zugriffs von mehreren Softwareentwicklern auf die Dateien.

6.5 Fazit

Die in diesem Kapitel vorgestellte Softwarearchitektur für die Verteilung der Daten ermöglicht die KoDeMat-Plattform Funktionen zur Integration und Überwachung von komplexer dezentral-gesteuerter Materialflusssysteme zu unterstützen. Ferner erlaubt es Änderungen an den verteilten Daten nachzuvollziehen. Anhand der implementierten homogenen Datenhaltung (mittels eines *in-memory* Datenbanksystems) können die Schnittstellenverwaltung als auch die Planungsumgebung mit 3D-Visualisierung in der Plattform integriert werden und um Funktionen zur Zusammenarbeit von verteilt arbeitenden Akteure erweitert werden. Im nächsten Kapitel wird das Planungswerkzeug mit 3D-Visualisierung vorgestellt.

7 Implementierung eines Planungswerkzeugs mit 3D-Visualisierung

Basierend auf der Datenverteilungsarchitektur (siehe Kapitel 6) bietet die KoDeMat-Plattform ein Planungswerkzeug mit 3D-Visualisierung. Das Softwarewerkzeug setzt eine virtuelle 3D-Umgebung für die kollaborative Planung und das Testen von heterogenen Materialflusssystemen an. Mittels dieses Softwarewerkzeugs besteht für die Akteure die Möglichkeit zur kollaborativen Anwendung von Planungsverfahren (siehe Kapitel 4). Folgende Funktionen sind durch das Planungswerkzeug unterstützt:

1. Groblayoutplanung: Kollaborative Planung eines Anlagelayouts.
2. Visualisierung: Eine 3D-Visualisierungsumgebung die mehrere Benutzern ermöglicht, das Modell eines Materialflusssystems zu betrachten. Die Zielsetzung für die Visualisierung ist die Abbildung des Zusammenspiels der Subsysteme in einer einheitlichen Anwendung.
3. Testen: Durchführung von Integrationstests zwischen verschiedenen Subsystemen eines gesamten Materialflusssystems.

Im Folgenden, werden die Funktionen die das KoDeMat-Planungswerkzeug unterstützt vorgestellt.

7.1 Kollaborative Planung und Erstellung eines Anlagelayouts

Das 3D-Hallenmodell als Ergebnis der vorausgehenden Phase der Vorarbeiten (siehe Kapitel 4) wird in die virtuelle Umgebung importiert. Ferner besteht die Möglichkeit einen Grundrissplan als Bilddatei (z. B. als .jpeg, .png) in die Umgebung zu importieren. Basierend auf dem als 3D-Objekt importierten Abbild der Halle, können die ersten Strukturvarianten, durch Verwendung der KoDeMat-Plattform in digitaler Form erzeugt werden. Somit kann jeder Benutzer visuelle 3D-Objekte importieren. Diese erlauben, existierende CAD-Dateien in die Kollaborationsplattform einzubinden. Durch die Nutzung von aktuellen

CAD-Austauschformaten (z. B. .3ds, .fbx, .kollada) können komplexe fördertechnische Systemkomponenten integriert werden. Dem Benutzer steht eine Reihe von digitalen Layoutbearbeitungswerkzeuge zur Verfügung, um ein Layoutelement in die Lagerhalle einzufügen. Um die 3D-Objekte im Raum anzuordnen und ein Layout zu erzeugen, unterstützt das Werkzeug eine synchrone Bearbeitung der Objekte mittels Editorfunktionen wie das Laden, Verschieben, Drehen und Löschen von Objekten. Die Benutzeroberfläche bietet Möglichkeiten für die Feinpositionierung eines 3D-Objektes im virtuellen Raum sowie eine *drag-and-drop* Funktion die es erlaubt die eindeutige Positionierung des Objektes herzustellen (siehe Abbildung 7.1). In der 3D-Umgebung kann der räumliche Platzbedarf der Materialflusssysteme überprüft werden. Derart werden beispielsweise die geplanten Systemkomponenten an ihrem vorgesehenen Ort platziert und potenzielle Kollisionen rechtzeitig erkannt. Durch die visuelle Darstellung von Funktionseinheiten (z. B. Förderstrecken, Kommissionierplätze) erhalten die beteiligten Akteure eine gute Wahrnehmung, die auch eine Kommunikationsbasis für Evaluierung zusätzlichen Layouts bietet.

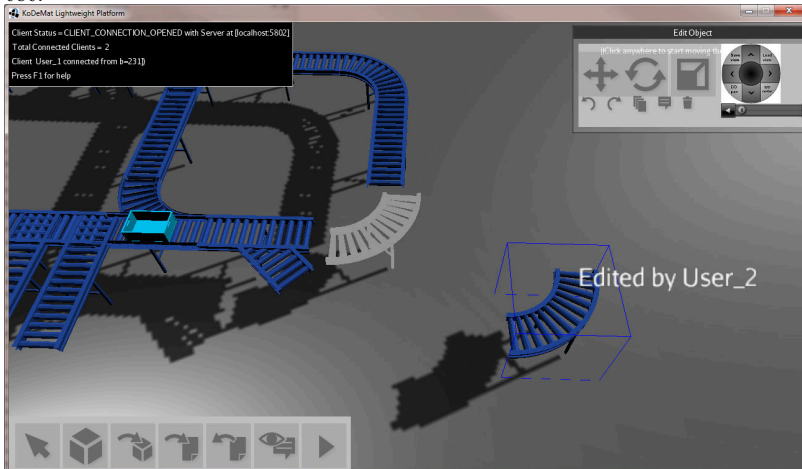
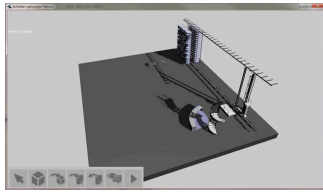
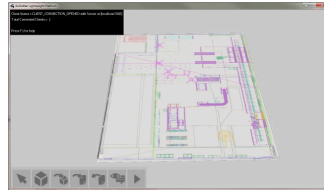


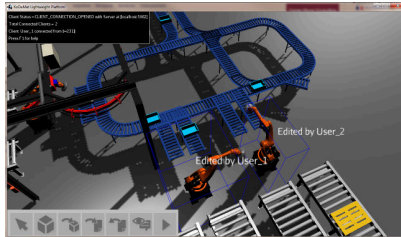
Abbildung 7.1: Positionierung eines 3D-Objekts in der virtuellen Umgebung des KoDeMat-Plattform



(a)



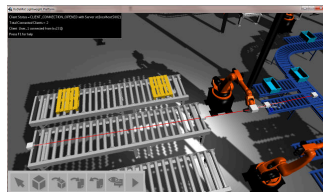
(b)



(c)



(d)



(e)



(f)

Abbildung 7.2: (a) CAD 3D-Modelle als 3D-Hallenlayout importieren; (b) Importierter Grundrissplan einer Lagerhalle; (c) Selektierte Objekte sind mit einem Rahmen markiert, der Name des Akteurs wird angezeigt; (d) Akteure können andere Teilnehmer auf interessante Punkte hinweisen; (e) Beziehungen zwischen Subsysteme können graphisch definiert und visualisiert werden (siehe auch Abschnitt 7.1); (f) visuelle Metapher eines Notizzettels

Eine wichtige Funktion des Werkzeugs ist die Möglichkeit neue Mitglieder in einer laufenden Sitzung einzuladen. Die entwickelte Datenverteilungsarchitektur ermöglicht dass die Daten weiterhin verfügbar bleiben und somit neu hinzugekommene Mitglieder Zugriff auf den aktuellen Stand gewährt. Die Kommunikation zwischen den Akteuren läuft, bis auf die verbale Kommunikation, innerhalb der KoDeMat-Plattform ab. Um eine Zusammenarbeit zwischen den beteiligten Benutzer zu unterstützen, werden spezielle interaktive Funktionen implementiert. Auf dieser Weise wird die Option angeboten, neben den fachlichen visuellen Modellen, Markierungen und Kommentare in der virtuellen Umgebung zu erzeugen. Abbildung 7.2(f) zeigt einen Screenshot der Demonstrationssoftware mit der visuellen Metapher eines Notizzettels in der Bildmitte. Der damit markierte Übergabepunkt zwischen zwei Rollenförderern bezeichnet gleichzeitig die Position einer Schnittstellendefinition (siehe Abschnitt 8). Am Ende einer Sitzung sollen sich alle beteiligten Akteure auf eine Lösung einigen, so dass ein Groblayout als Ergebnis dieser Planungsphase feststeht.

7.2 Planung virtueller Integrationstests

Die KoDeMat-Plattform erlaubt die Prüfung des Zusammenspiels der Anlagensteuerung bei einer virtuellen Inbetriebnahme (siehe [USD11, S. 215]) durch das digitale Anlagenmodell. Dabei werden dieselben agentenbasierten Steuerungen verwendet, die später auch in der realen Anlage zum Einsatz kommen. Das bietet dem Entwickler von Steuerungsprogrammen die Möglichkeit vor dem Aufbau die Anlage zu testen (vgl. [USD11, S. 303]). Weiterhin ermöglicht die Softwareplattform, die Durchführung virtueller Integrationstests. Für die Durchführung von Integrationstest zwischen Subsystemen erfolgt zur Abbildung der benötigten Materialflussinformationen eine Modellierung der Anlagentopologie, woraufhin Test-Transportaufträge spezifiziert werden können.

Definition der Topologie

Im ersten Schritt der Modellierung eines Materialflusssystemes gilt es die Topologie, also die Nachbarschaftsbeziehungen zwischen den einzelnen Systemkomponenten, zu definieren. Hierzu ist zunächst eine anforderungsgerechte Modellierung des Gesamtsystems erforderlich, wofür verschiedene Methoden zur Auswahl stehen (siehe Abbildung 7.3).

Aufgrund der auch bei komplexen Systemen gegebenen Übersichtlichkeit sowie dem vorliegenden Modellierungszweck, also der Ableitung einer Netzwerktopologie, erfolgt die Modellierung des Gesamtsystems in dieser Arbeit mittels Graphen [Gü10]. Im Folgenden werden daher die wesentlichen Aspekte der Graphentheorie kurz beschrieben.

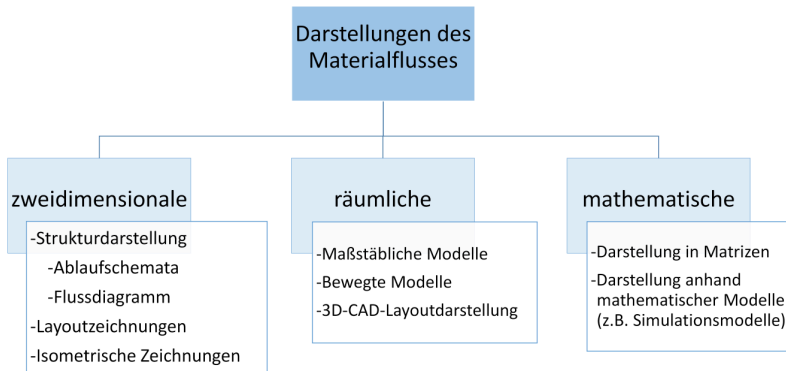


Abbildung 7.3: Darstellungsmethoden für die Abbildung des Materialflusses (vgl. [AF09, S. 348], [Kam13, S. 13])

Zunächst besteht ein Graph aus einer Menge von Knoten, die über Kanten miteinander verbunden sind. Werden die Kanten durch Pfeile, die eine Bezugsrichtung angeben, visualisiert, spricht man von gerichteten Graphen (Digraphen) (siehe Abbildung 7.4). Somit lassen sich Beziehungen, Abläufe sowie Verzweigungen darstellen (siehe Abbildung 7.6(e)). Gerichtete Graphen können folglich als Basis für die Modellierung des Materialflusssystemes in der vorliegenden Arbeit betrachtet werden. Neben der bereits beschriebenen Modellierung von Prozessen, sieht die Graphentheorie zudem die Abbildung der Struktur eines Systems vor. Da das untersuchte System hierarchisch aufgebaut ist (siehe Abschnitt 4.1), empfiehlt sich in diesem Kontext der Einsatz hierarchischer Graphen [Meb08, S. 38]. Damit können die Beziehungen sowohl zwischen Subsystemen als auch zwischen Systemkomponenten dargestellt werden. In Abbildung 7.5 ist ein System mittels eines hierarchischen Graphen modelliert, wobei die Knoten A

bis E Subsysteme darstellen. Die Knoten der Unternetzwerke, in die ein Subsystem unterteilt werden kann, entsprechen den verschiedenen Systemkomponenten.

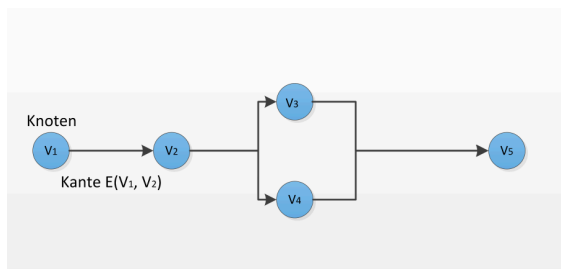


Abbildung 7.4: Schema eines gerichteten Graphen

Relevante Definitionen in Bezug auf Graphen sind (vgl. [Meb08]):

- Ein ungerichteter Graph G besteht aus einem geordneten Mengenpaar $G = (V, E)$. Die Elemente von V heißen Knoten, und die Menge E besteht aus zweielementigen Teilmengen $x, y \in V$, die als Kanten bezeichnet werden. Wir schreiben kurz $n(G) = |V|$ und $m = m(G) = |E|$ für deren jeweilige Anzahl.
- Bei gerichteten Graphen (Digraph) sind die Teilmengen x, y durch geordnete Paare (x, y) zu ersetzen.
- Die Kante x, y verbindet die Endpunkte $x \in V$ und $y \in V$.
- Sind x und y aus V durch eine Kante miteinander verbunden, so heißen sie adjazent oder benachbart.

Um das steuerungslogische Modell eines Materialflusssystems mittels Graphen zu erstellen, besteht die Möglichkeit 3D-Modelle von Fördertechnikmodulen in die virtuelle 3D-Umgebung der Plattform zu importieren. Zu jeder Systemkomponente können Ein- und Ausgänge definiert werden, welche untereinander verbunden werden können. Die Verbindungspunkte werden in dieser Arbeit als Anschlusspunkt (AP) bezeichnet und bilden die Knoten des Netzwerks. Die Verbindungen (Kanten) zwischen Anschlusspunkten definieren die Übergänge von

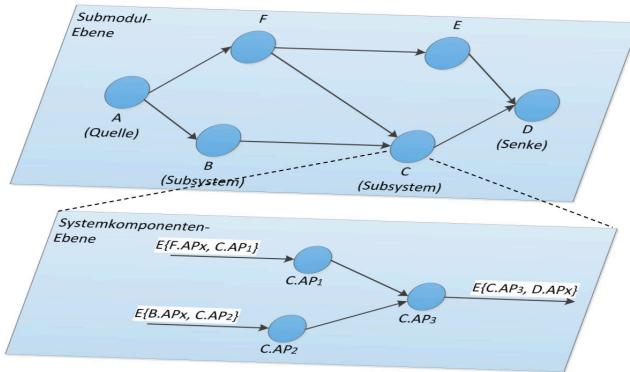


Abbildung 7.5: Darstellung eines logistischen Netzwerks mittels eines hierarchischen Graphen

einem Subsystem zum anderen. Diejenigen Kanten, die Anschlusspunkte innerhalb eines Subsystems verbinden, werden interne Kanten genannt (siehe Abbildung 7.6 (a)). Verbindungen zwischen unterschiedlichen Subsystemen werden als externe Kanten bezeichnet (siehe Abbildung 7.6(c)). Auf die gleiche Weise ist es möglich die Verbindungen eines Verzweigungsmoduls zu modellieren (Abbildung 7.6(f)). Zur virtuellen Modellierung eines komplexen Materialflusssystems bietet die KoDeMat-Plattform die Funktionen eines Multi-User-3D-Planungswerkzeuges, mit welchem die Verbindungen zwischen Systemkomponenten definiert werden können. Wie auf den Screenshots der 3D-Visualisierungsumgebung der KoDeMat-Plattform dargestellt ist (siehe Abbildung 7.6(b),(d) und (f)) kann der Anwender die Anschlusspunkte als 3D-Objekte definieren und so die Beziehungen festlegen. Auf diese Weise kann der Anwender die Topologie eines gesamten Materialflusssystems erzeugen (vgl. Abbildung 6.11). Wie in Abschnitt 6.4 beschrieben, kann die Topologie daraufhin exportiert und mittels externer Softwarewerkzeuge als Graph und Materialflussmatrix visualisiert werden.

Es können ein oder mehrere Testfälle definiert werden. Ein Testfall besteht aus folgenden Datenstrukturen:

- Anlagelayout

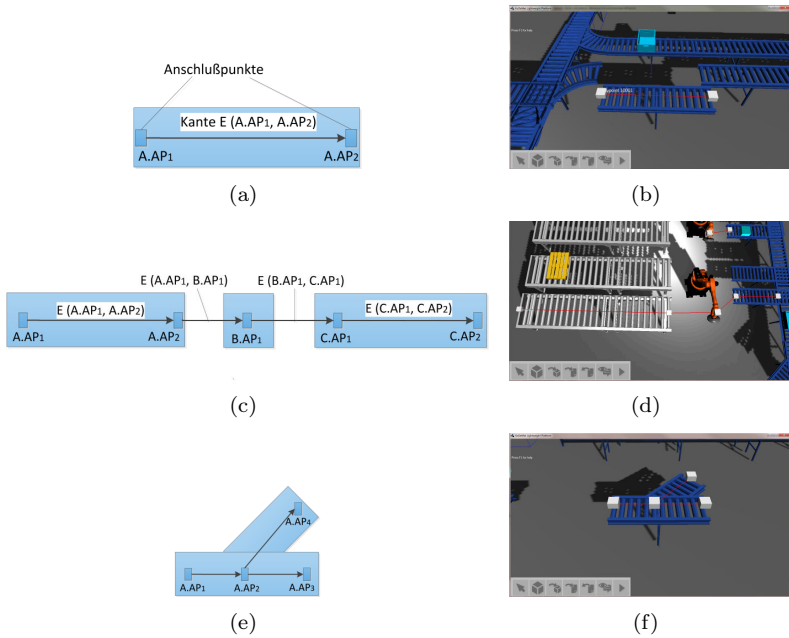


Abbildung 7.6: (a) Anschlusspunkte verbunden mit internen Kanten innerhalb einer Systemkomponente; (b) 3D-Modell einer Systemkomponente mit definierten Anschlusspunkten in der 3D-Umgebung von KoDeMat-Plattform; (c) Anschlusspunkte mit externen Kanten zwischen unterschiedlichen Subsystemen; (d) 3D-Modell unterschiedlicher Subsystemen mit definierten Anschlusspunkten in der 3D-Umgebung von KoDeMat-Plattform; (e) Interne Verbindungen eines Verzweigungssystems (f) 3D-Modell eines Verzweigungssystems mit definierten Anschlusspunkte in der 3D-Umgebung von KoDeMat-Plattform;

- Quellen und Senken
- Dienstorte (z. B. Kommissionierplatz)
- Lastdaten (z. B. Transportaufträge)
- Voraussetzungen zur Durchführung (z.B. Subsysteme vorhan-

den)

- Erwartete Ergebnisse zur Validierung (z. B. Transportauftrag erreicht bestimmte Senke)

Während der Durchführung eines Testlaufs fungiert die Testumgebung als Koordinator von Ereignissen und ist in diesem Sinne vergleichbar mit einem Simulations- bzw. Emulationswerkzeug. Die Durchführung der Lastbewegungen und die Steuerung des Systems obliegen jedoch den angeschlossenen Subsystemen. Eine Validierung erfolgt durch den Abgleich der erwarteten Ergebnisse, die in Form von Ereignisdefinitionen vorliegen, mit den während des Testlaufs erhaltenen Ereignissen. Die erwarteten Ereignisdefinitionen werden im Fehlerfall visualisiert. Grundsätzlich werden die –von den Systemkomponenten–Agenten– generierten Ereignisse (siehe Abbildung 7.7, die während eines Testlauf auftreten, aufgezeichnet und über die Benutzeroberfläche anhand verteilte Tabellen–Datenstrukturen (siehe Abschnitt 8.3) zugänglich gemacht, so dass die Experten des Herstellers im Fehlerfall einen Zugriff auf alle relevanten Informationen erhalten.

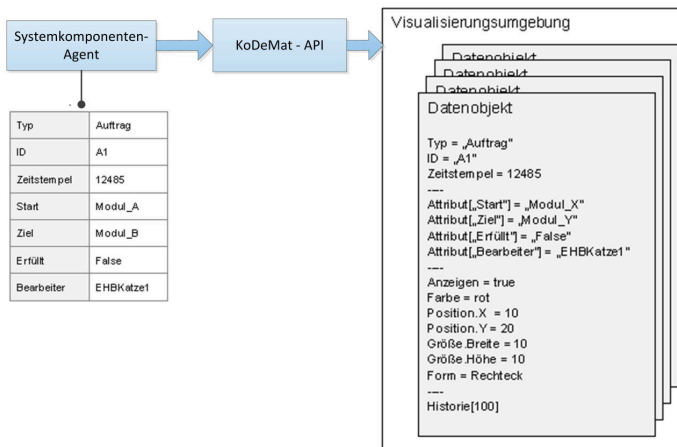


Abbildung 7.7: Visualisierung von agenten-generierten Ereignissen (vgl. [Chi10, S. 122])

7.3 Visualisierung realer Materialflusssystemen

Die Visualisierung und Überwachung einzelner Systeme stellt keine besondere Herausforderung dar [Chi10, S. 121]. Heutzutage bieten die Mehrheit der Hersteller von Förderanlagen eine Visualisierungsfunktion für ihre Systeme an, die oft in Form eines Webdienstes realisiert ist. Weiterhin gibt es am Markt verschiedene Softwarelösungen, die 2D- und 3D-Visualisierungsfunktionen für Materialflusssysteme und Anlagen bieten. Beispiele für solche Softwarelösungen sind Demo3d ([Emu14]) und Plant Simulation ([Sie14]). Diese Softwarelösungen sind jedoch lediglich auf herkömmliche, hierarchisch strukturierte und zentralgesteuerte Systeme ausgerichtet.

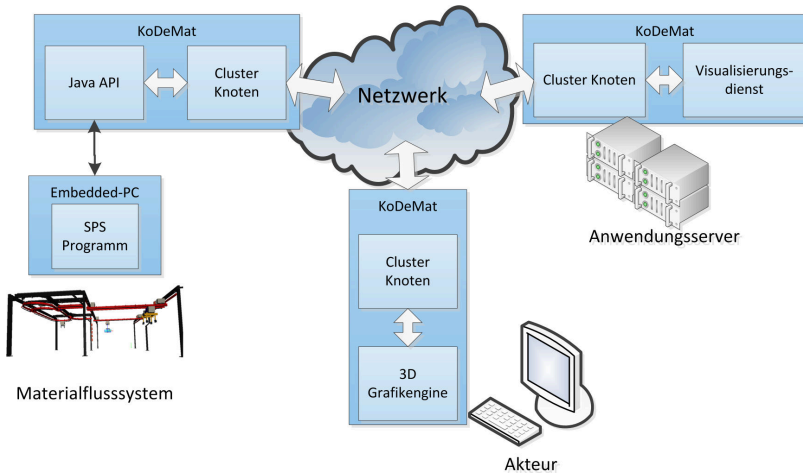


Abbildung 7.8: Netzwerkarchitektur zur Visualisierung realer Systeme

Dezentral gesteuerte Materialflusssysteme stellen die Herausforderung dar, dass bei solchen Systemen die Visualisierungssoftware einen Zusammenhang von Informationen aus einem verteilten Netzwerk autonomer Einheiten rechtzeitig liefern muss. Darüber hinaus entstehen durch die in dieser Arbeit betrachteten heterogenen Systeme weitere Herausforderungen, da sie auf verschiedenen Steuerungskonzepten basieren. Das bedeutet, dass jedes System einen anderen Mechanismus benutzt, um die Agentenmeldungen zu verteilen (siehe Peer-to-peer, Push,

Pull Prinzipien (siehe [Chi10, S. 69]). Die relevanten Informationen von den Agenten (wie z.B. Topologie und Statusmeldungen) müssen synchronisiert, zeitlich geordnet und versioniert (durch eine Änderungshistorie) werden. Die Visualisierungsumgebung ist so gestaltet, dass die beschriebenen Visualisierungsfunktionen gleichzeitig bei mehrere entfernte Akteure zugreifbar sind. Weiterhin es ist möglich vielen unterschiedlichen Materialflussanlagen zu visualisieren, ohne dass dazu eine Neuprogrammierung erforderlich ist.

Die entwickelte Softwareplattform bietet eine programmierbare Schnittstelle (Application Programming Interface (API)), die es ermöglicht, eine einfache Anbindung externer Systeme zu erstellen. Die KoDeMat-Plattform ist in der Lage als Ausgangspunkt für die generierte Status-Meldungen der Systeme zu dienen. Die Datentransferprotokolle werden in die Schnittstellenverwaltung integriert und somit für externe Programme verfügbar gemacht. Dadurch ist die Verwendung mehrere Visualisierungssysteme, sowie VR-fähige Ausgabegeräte möglich (siehe Abschnitt 7.3).

Visualisierung von Flurförderzeugen

Im Rahmen der Forschungsarbeiten erfolgt der Einsatz der KoDeMat-Plattform als Visualisierungswerkzeug zur Positionsanzeige von einem. Im Rahmen des vom Lehrstuhl fml durchgeführten Forschungsprojektes *Das Staplerauge* wurde auf Basis einer Kamera verschiedene Sensorfunktionen umgesetzt. Durch Methoden der Echtzeit-Bildauswertung wurden Sensorfunktionen zur Erkennung von Zuständen des Flurförderzeugs und Merkmalen von Waren entwickelt [HG12]. Das im DSA-Projekt entwickelte System erfüllt Funktionen wie Positionsbestimmung des Flurförderzeugs im Raum und eine Hubhöhen Messung (siehe [JHG13]):

Die KoDeMat-Plattform dient als ein Multi-User Visualisierungswerkzeug für die oben genannten Sensordaten. Im einen Testszenario wird die kamerabasierte Fahrzeuglokalisierung verwendet um die Position eines Gabelstaplers in der Versuchshalle des Lehrstuhls fml zu bestimmen. Die Positionsbestimmung erfolgt anhand 2D-Markern die unter der Hallendecke aufgebracht sind. Nach Ermittlung der Sensordaten, sendet das System die aktuelle Position des Fahrzeugs per WLAN an den KoDeMat-Visualisierungsserver. Darüber hinaus werden noch

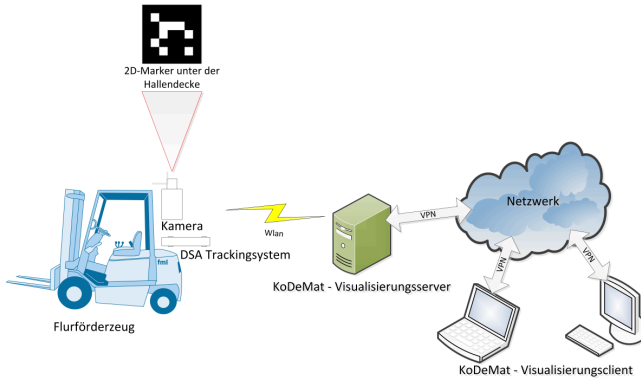


Abbildung 7.9: Einsatzszenario der KoDeMat-Plattform als Visualisierungswerkzeug für Flurförderzeuge

weitere Zustandsdaten wie die Hubhöhe ermittelt. Die Datenübermittlung findet durch UDP-Datagramme (User Datagram Protocol (UDP)) statt. Die Koordinaten werden von dem Visualisierungswerkzeug in einem lokalen Koordinaten-System transformiert. Anschließend kann die Position des zugehörigen 3D-Objektes aktualisiert werden. Abbildung 7.3 illustriert dieses Szenario und Abbildung 7.3 zeigt die Kameraaufnahme und die Visualisierung des Gabelstaplers im 3D-Umgebung der KoDeMat-Plattform.

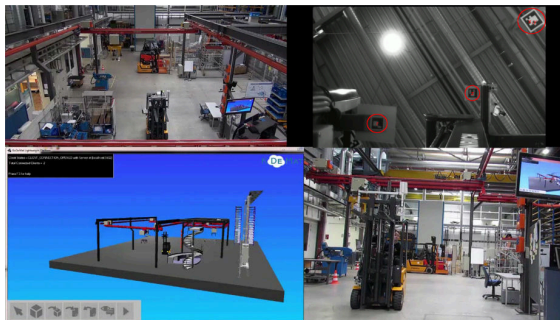


Abbildung 7.10: Visualisierung eines Flurförderzeugs mittels der KoDeMat-Plattform

Plattformübergreifende Kollaboration

Die modulare Softwarearchitektur ermöglicht es, dass innerhalb der KoDeMat-Plattform mehrere Benutzer gleichzeitig in derselben virtuellen Umgebung agieren können. Dabei müssen die Anwender nicht zwingend das gleiche Visualisierungssystem (z. B. 3D-Grafikengine) verwenden. Anhand der entwickelten Softwarearchitektur (siehe 6.1 und der Plattform API (siehe Abbildung 7.8) besteht die Möglichkeit verschiedene Visualisierungssysteme und Interaktionsgeräte mit der KoDeMat-Plattform zu koppeln und als Kollaborationshilfsmittel zu nutzen. Die nächsten Abschnitte stellen zwei solche Anwendungsfälle vor.

Unterstützung vom Virtual-Reality

Unter Berücksichtigung der bereits erfolgreichen Anwendung der Virtual-Reality (Virtual Reality (VR)) im Bereich Planung und Visualisierung in der Logistik (siehe [Kam13]) und [USD11]) wurde ein Ansatz zur Unterstützung des KoDeMat Planungswerkzeugs mittels VR-Technologie entwickelt und in die Plattform integriert. Basierend auf der Datenverteilungsschicht (siehe Abschnitt 6.2), wurde ein Visualisierungsklient für eine zweite Grafikengine implementiert. Diese Grafikengine namens Unity3D unterstützt eine Integration der Oculus Rift VR-Brille [Ocu14] mittels eines Plug-Ins. Damit wird es möglich das VR-Ausgabegerät mit der KoDeMat Visualisierungsumgebung zu koppeln.

Im Gegensatz zur 3D-Visualisierung stellt die VR-Technologie eine neuartige Mensch-Maschine-Schnittstelle dar, die zur Visualisierung von 3D-Modellen dient und die Manipulation dieser mit geeigneten Interaktionstechniken ermöglicht ([Wie01, S. 139]). Dadurch steigen der Grad der Immersion und die Integrationsfähigkeit der Benutzer. Nach Hausstädler [Hau08] ist unter Immersion die Integration des Benutzers in die virtuelle Welt zu verstehen. Die Interaktion beschreibt die Fähigkeit der Nutzer die digitale Welt aktiv zu beeinflussen und so mit der immersiven virtuellen Umgebung zu interagieren. Die entwickelte 3D-Benutzeroberfläche (siehe Abbildung 7.11) ermöglicht im kollaborativen Planungsprozess die Lagerhallen in der VR zu visualisieren und zu manipulieren. Das System nützt die VR-Brille von

Firma Oculus. Der Einsatz einer VR-Brille im Rahmen der Logistikplanung ist besonders vorteilhaft, da eine große virtuelle Umgebung erstellt wird. Die VR-Brille hilft dabei Dimensionen besser abzuschätzen, mögliche Ungereimtheiten in der Planung zu finden und den Benutzer stärker ohne die Ablenkungen von der Außenwelt einzubinden. Weiterhin ermöglicht die VR eine virtuelle Begehung was auch für Marketingzwecke eingesetzt werden kann.



Abbildung 7.11: Die 3D-Visualisierungsumgebung von KoDeMat-Plattform wurde erweitert um VR-Brille von Oculus [Ocu14] zu unterstützen

Unterstützung eines Multi-Touch-Screen

Zusätzlich zu der Integration der VR-Brille, wird eine Erweiterung der Visualisierung implementiert, die einen Multi-touch-Screen als Interaktionsgerät unterstützt (siehe Abbildung 7.12). Multi-Touch-Screens sind Monitore die eine berührungsempfindliche Oberfläche für die Ein-

gabe von Daten mit Hilfe von Gesten ermöglichen. Der kapazitive oder optische Touch-Screen erkennt gleichzeitig mehrere Berührungen. Er kann horizontal wie ein Tisch aufgestellt werden und erlaubt mehreren Benutzer interaktiv an der Erstellung eines Layouts zu arbeiten. Die Oberfläche des Visualisierungswerkzeugs wird um intuitive touchfähige Funktionen erweitert. Die Touch-Funktionalität ist bei mobilen Geräten weitverbreitet und nutzt dafür Funktionen wie die gestenbasierte Kamerasteuerung. Die Eignung und die Vorteile des Einsatzes eines Planungstables kombiniert mit der 3D-Darstellung wurde in Planungsverfahren bereits erfolgreich getestet (siehe [EKTW12]).



Abbildung 7.12: Einsatz eines Multi-Touch-Screens als Interaktionsgerät für das Visualisierungswerkzeug

7.4 Technische Herausforderungen bei der synchronen kollaborativen Bearbeitung

Die Komplexität kollaborativer Echtzeiteditoren entsteht aufgrund der Latenzzeit zwischen den Kommunikationsvorgängen. Wenn die Kommunikation in idealisierter Weise unverzögert stattfände, dann wäre die Erstellung eines kollaborativen Echtzeiteditors nicht schwie-

riger, als bei einem einfachen Editor für ein Single-User-System. Jedoch ist die Kommunikationsgeschwindigkeit durch die Latenz des Netzwerks beschränkt. Die Akteure ändern gleichzeitig ein Dokument, welches jeweils lokal zwischengespeichert wird. Einerseits werden die eigenen Änderungen lokal sofort angezeigt, andererseits werden die externen Änderungen zeitverzögert übertragen. Dies erzeugt auf allen Systemen Inkonsistenzen in den lokalen Versionen, die aufgelöst werden müssen.

So entsteht eine technische Herausforderung für kollaborative Echtzeiteditoren, die externen Änderungen (durch entfernte Akteure) in ein lokales Dokument zuintegrieren, so dass sie nicht mit lokalen Änderungen in Konflikt geraten. Hinzu kommt, dass alle lokalen Dokumente am Ende den gleichen Inhalt darstellen sollten. Demnach müssen die Änderungen aller Benutzer wechselseitig in den jeweiligen lokalen Dokumenten konsistent integriert werden. Einige Ansätze verwenden das Sperren von Dokumenten, so dass nur jeweils ein Akteur Änderungen durchführen kann, oder lösen das Problem durch einen manuell durchzuführenden Verschmelzungsvorgang. Beide Ansätze sind jedoch für die Akteure ungünstig, da ein gemeinsames Arbeiten so nur noch eingeschränkt möglich ist bzw. umständlich wird.

Anspruchsvollere Lösungen verwenden eine Technik, die kein Sperren von Dokumenten verwendet und die beliebig viele Akteure zulässt, so dass das Dokument in allen Bereichen frei veränderlich ist. Dies kann beispielsweise durch eine Client-Server-Architektur ermöglicht werden, bei welcher der Server die Synchronisation der Änderungen auf allen Clients übernimmt und dabei auch die Auswirkungen der Latenzzeit berücksichtigt. Dies kann entweder über eine serverseitige Zeitsynchronisation geschehen, bei der die Clients die Änderungen mit einem Zeitstempel versehen und an den Server schicken, der sie zeitlich korrekt sortiert an die restlichen Clients verbreitet. Ein anderer Ansatz, wie er auch in der KoDeMat-Plattform zum Einsatz kommt, lässt alle Änderungen zentral über den Server laufen, der diese in der Reihenfolge ihres Eintreffens erneut an die Clients verteilt, so dass eine globale Ordnung garantiert werden kann. Die lokalen Änderungen werden so erst angezeigt, wenn diese vom Server freigegeben werden. Dieser einfache Ansatz besitzt große Vorteile, wenn die Anzahl an Akteuren begrenzt und die Anforderungen an die Echtzeitdarstellung von lokalen Änderungen gering sind. Um eine effektivere Zusammenarbeit

zu ermöglichen, wird jedes Objekt das von einem Akteur selektiert ist und sich damit im Bearbeitungsmodus befindet, markiert und ein Schilder angezeigt, auf dem der Name des aktuellen Bearbeiters steht (siehe Abbildung 7.2(c)).

8 Implementierung der Schnittstellenverwaltung

Die Schnittstellenverwaltung für die KoDeMat-Plattform ermöglicht die Konzeptionierung, den Test sowie die virtuelle Inbetriebnahme komplexer Intralogistiksysteme, an deren Erstellung mehrere Hersteller beteiligt sind, indem der Austausch relevanter Informationen über Systemgrenzen hinweg ermöglicht wird. Sie basiert auf einem ähnlichen Datenmodell, wie es die Visualisierung verwendet, wobei die Datencontainer hier Kommunikationsobjekte beschreiben.

Für die Kommunikation zwischen den Systemen spielt es keine Rolle, ob Daten von Steuerungsinstanzen herkömmlicher, hierarchisch strukturierter Anlagen oder direkt von dezentral gesteuerten Modulen gesendet bzw. empfangen werden. Somit besteht für jedes Subsystem Wahlfreiheit bzgl. des eingesetzten Steuerungsparadigmas. Dabei gilt es zu beachten, dass sich dezentral gesteuerte Systeme aufgrund ihres stark modularen Aufbaus, des hohen Anteils an wiederverwendbarer Software sowie spezifizierter Kommunikationsschnittstellen in besonderem Maße für die hier angestrebte Form des Engineerings eignen.

Im Wesentlichen befasst sich die Schnittstellenverwaltung der KoDeMat-Plattform mit Kommunikationseinstellungen und der Definition von verschiedenen Telegrammformen sowie deren Kodierung. Die Spezifikation kann um Transformationsfunktionen oder Zustandsautomaten erweitert werden, die bestimmte Inhalte umwandeln beziehungsweise Protokolle implementieren. Die Schnittstellenverwaltung wird als Werkzeug zur reibungslosen Zusammenarbeit zwischen den Subsystemen der verteilten Materialflussteuerung eingesetzt.

Für die Entwicklung der Schnittstellenverwaltung wird zur Abbildung der benötigten Materialflussinformationen anschließend eine Modellierung der Anlagentopologie durchgeführt, woraufhin die Schnittstellen des Systems spezifiziert werden können. Aufbauend auf diesen Ergebnissen kann ein kollaborativer Planungsprozess für Schnittstellen entwickelt und auf das Gesamtmodell angewendet werden.

8.1 Agentenorientierte Konfiguration von Schnittstellen zwischen Subsystemen

Für die Konfiguration der Schnittstellen der Subsysteme wurde ein Ansatz gewählt, der von einer agentenorientierten Sichtweise auf das Gesamtsystem ausgeht. Jedes Subsystem, unabhängig von seiner internen Steuerungsstruktur, wird als Agent betrachtet und verfügt damit über genau spezifizierte Kommunikationsschnittstellen. Für einen solchen Einsatz von autonomen Steuerungsagenten ohne eine zentrale Steuerungsinstanz sind bereits Vorgehensweisen für den Entwurf geeigneter Kommunikationsmodelle entwickelt worden [HLR11], auf die im Rahmen dieses Projekts zurückgegriffen wird. Im Bereich der Multiagentensysteme wird der Interoperabilitätsgrad durch Standards unterstützt, wie z.B. FIPA [FIPc], Mobile Agent Facility Specification [OMG13] oder die Open Agent Architecture. Das weitere Vorgehen orientiert sich am FIPA-Standard, in dem die folgenden drei Ebenen für die Agentenkommunikation spezifiziert werden: eine Ontologie, Kommunikationsprotokolle und die Sprache.

Die Ontologie bildet das gemeinsame Datenmodell aller Subsysteme, so dass eine gemeinsame Semantik des Problemfeldes der Materialflusststeuerung ermöglicht wird. Eine Grundvoraussetzung der Integration heterogener Subsysteme ist die Schaffung einer solchen gemeinsamen Ontologie, in dem z.B. das Vorhandensein von Transportaufträgen, Routinganfragen oder Lastübergabeprotokollen von allen Subsystemen prinzipiell bereits unterstützt wird (oder neu entwickelt werden muss). Auch bestimmte Verhaltensweisen von Systemen im Rahmen einer Lastübergabe, wie Freigabe-Handshakes, sind Bestandteil dieser Ontologie. Die KoDeMat-Plattform unterstützt mit ihrer Schnittstellenverwaltung die Integration nicht-standardisierter Kommunikationsschnittstellen auf Basis einer gemeinsamen Ontologie, indem sie eine Möglichkeit bietet die Kommunikationsprotokolle, sowie die Struktur und Kodierung der Nachrichten kollaborativ zu entwickeln.

Kommunikationsprotokolle fokussieren den Informationsaustausch zwischen den Agenten und beschreiben die Struktur der Nachrichten [FIPa]. Von der FIPA ist eine Agent Communication Language (ACL) standardisiert [FIPa], deren Architektur als Basis für die Schnittstellenverwaltung der KoDeMat-Plattform dient. Eine in dieser Hinsicht

bedeutende Eigenschaft der ACL ist die Unabhängigkeit der Nachrichteninhalte von ihrer Struktur und Kodierung. Die grundsätzliche Annahme, dass FIPA-ACL-Nachrichten aus einer Liste von Nachrichtenparametern bestehen, wurde für die KoDeMat-Plattform übernommen. Die Inhalte von Nachrichten bestehen daher aus Listen von Attributen, die jeweils einen Namen und einen Wert besitzen.

Für eine erfolgreiche Interaktion zwischen den Subsystemen muss im Sinne der Ontologie die Reihenfolge und der jeweilige Typ einer Nachricht in Form von Protokollen festgelegt werden. In Abbildung 8.1 ist ein solcher Nachrichtenaustausch als Sequenzdiagramm des FIPA Request Interaction Protocol dargestellt (vgl. [FIPb]). Innerhalb dieser Abbildung ist auch ein Beispiel einer FIPA-ACL-Nachricht mit einem sogenannten REQUEST-Sprechakt dargestellt. Dieses Protokoll wird für eine Interaktion verwendet, bei der ein Agent (Initiator) einen anderen Agenten zu einer Aktion auffordert. So eine Aktion könnte z. B. die Anfrage zur Initialisierung eines Lastübergabeprozesses sein. Dieser Agent (hier *Participant* genannt) hat die Möglichkeit, die Aufforderung abzulehnen (*refuse*) oder ihr zuzustimmen (*agree*). Am Ende wird der Initiator-Agent über das Ergebnis der Aufforderung mit einer entsprechenden Meldung (Erfolg oder Fehler) informiert (vgl. [HLR11]). In der KoDeMat-Plattform können diese Protokolldefinitionen erstellt und bearbeitet werden.

Die Protokolle beschreiben nur den Kommunikationsvorgang an einer Schnittstelle, sie stellen jedoch nicht die Datenstruktur oder den Inhalt der Nachrichten selbst dar. Um die Verständigung der Kommunikationsteilnehmer zu ermöglichen, muss eine computerlesbare Sprache existieren, welche die in den Nachrichten codierte Semantik beschreibt (vgl. [HLR11]). Diese Sprachdefinition muss den jeweiligen Subsystemen bekannt sein, damit diese empfangene Nachrichten verarbeiten können. In den folgenden Abschnitten wird die KoDeMat-Schnittstellenverwaltung beschrieben, mit der dieser Arbeitsschritt kollaborativ möglich ist.

8.2 Schnittstellenverwaltung als temporäre Agentenkommunikationsplattform

Damit eine agentenübergreifende Kommunikation in der Test- und Planungsphase eines Materialflusssystemes möglich wird, handelt die

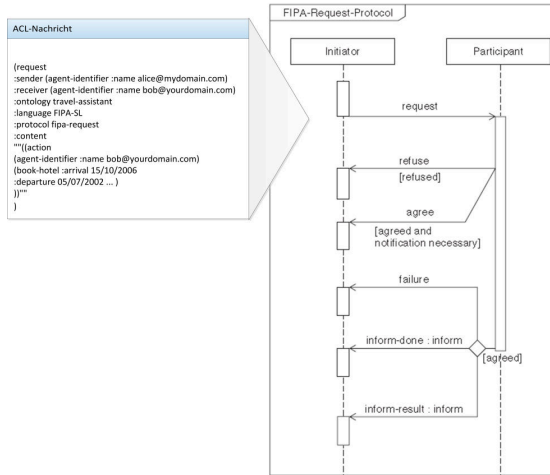


Abbildung 8.1: Sequenzdiagramm des FIPA Request Interaction Protocol [FIPb]

Schnittstellenverwaltung zur Laufzeit als Agentenplattform. Sie ist in der Planungsphase ein Element der Systemarchitektur, über das die Subsysteme untereinander und mit der Umwelt interagieren können. Eine Besonderheit der Agentenplattform ist die Implementierung eines Schnittstellenagenten, der die Rolle einer Kommunikationsbrücke zwischen unterschiedlichen Agentenplattformen übernimmt. In Abbildung 8.2 sind zwei Subsysteme dargestellt. In jeder subsystemspezifischen Agentenplattform werden Agenten eingesetzt, die für die Steuerung der entsprechenden Systemkomponenten zuständig sind. Die KoDeMat-Plattform ermöglicht, die Schnittstellenagenten kollaborativ in einer Sitzung zu konfigurieren, damit zwei Subsysteme miteinander zusammenarbeiten können.

8.3 Kollaborative Schnittstellendefinition

Um den Schnittstellenverwaltungsprozess zu unterstützen, bietet die KoDeMat-Plattform einen kollaborativen Echtzeiteditor, der eine synchrone Bearbeitung von Schnittstellenspezifikationen in textueller Form ermöglicht. Der Editor ist in das Visualisierungsmodul integriert und

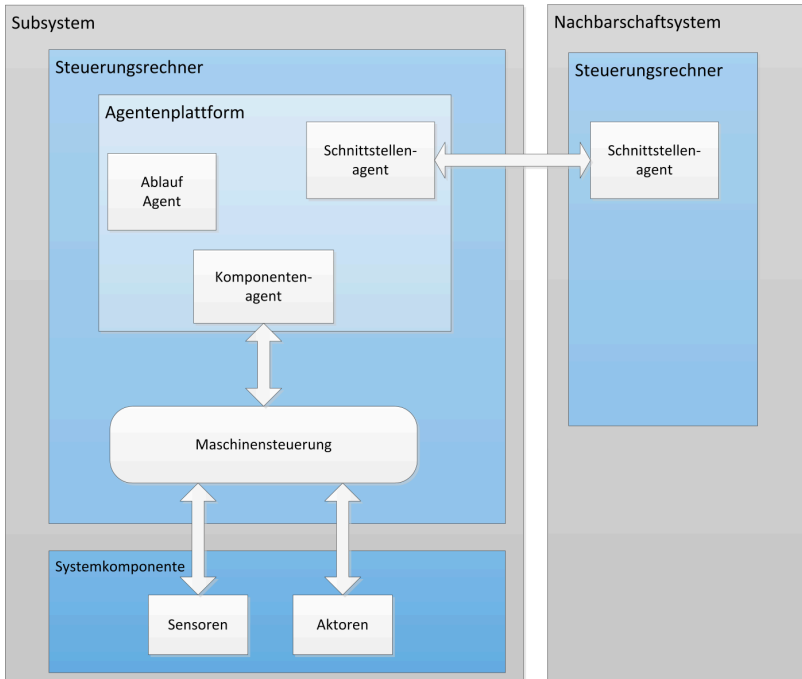


Abbildung 8.2: Kommunikation zwischen zwei agentenorientierten Materialflusssystemen mittels Schnittstellenagenten (vgl. [GHM10])

erlaubt die Erstellung von Nachrichten für die Schnittstellenagenten. Für die Darstellung und Bearbeitung von Agentennachrichten werden Tabellen verwendet, in welche die Nachrichtenparameter eingetragen werden. Der Inhalt der Tabellen wird zwischen den Akteuren synchronisiert, so dass jeder Benutzer in Echtzeit ein aktuelles Abbild der Schnittstellendefinition erhält. In einer separaten Spalte dieser Tabelle wird zusätzlich der Name des Benutzers, der aktuell einen spezifischen Parameter bearbeitet, angezeigt. Auf diese Weise erfolgt eine koordinierte synchrone Bearbeitung der Daten. Abbildung 8.3 zeigt einen Bearbeitungsvorgang mit zwei beteiligten Benutzern, die synchron an der Definition einer Agentennachricht arbeiten. Der vorgestellte Ansatz zur kollaborativen Schnittstellenverwaltung ist, wie bereits beschrieben, nicht nur auf dezentrale Materialflusssysteme

beschränkt, sondern kann auch für zentral gesteuerte Systeme eingesetzt werden.

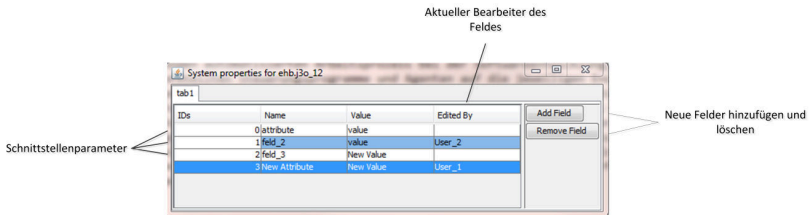


Abbildung 8.3: Screenshot des Echtzeiteditors für Nachrichtendefinitionen in der KoDeMat-Plattform. Mehrere Benutzer können gleichzeitig die Schnittstellenspezifikationen synchron definieren.

8.4 Automatische Generierung von Agentenkonfigurationsdateien

Die Schnittstellenverwaltung ermöglicht einen weitestgehenden automatisierten Arbeitsprozess in der Softwareentwicklung, da diese zur Codegenerierung genutzt werden kann. Die Funktion zur Codegenerierung erlaubt es, Änderungen an der Spezifikation einzelner Schnittstellen direkt in einen lauffähigen Code einzubauen. In Abbildung 8.4 wird an einem Beispiel dargestellt, wie anhand der erstellten Parameter die resultierende Nachricht als XML-Code visualisiert werden kann. Jede Aktualisierung der Parameter wird in den Code übernommen. Wenn zwischen den Projektpartnern eine Einigung bezüglich einer Spezifikation erreicht ist, können diese Agentenkonfigurationsdateien und alle Informationen zum Systemlayout abgespeichert und im Anschluss als lauffähige Schnittstelle in Betrieb genommen werden.

8.5 Fazit

Zusammenfassend lässt sich festhalten, dass die Schnittstellenverwaltung eine effiziente Projektierung heterogener Fördertechnikanlagen ermöglicht. Die logistische Funktionalität des Gesamtsystems bleibt dabei unabhängig von den auf Subsystemebene angewandten Steuerungskonzepten (zentral/dezentral) erhalten. Die Schnittstellenverwal-

9 Kopplung existierender Demonstratoren

Mit der KoDeMat-Plattform wurde ein Demonstrationsszenario entwickelt und implementiert, welches einen realen Einsatz im Projektablauf abbildet. Der Grundgedanke hinter diesem Vorgehen besteht darin, dass die beiden beteiligten Forschungsstellen ihre existierenden Demonstratoren für agentengesteuerte Materialflusssysteme im Rahmen eines fiktiven Großprojekts koppeln und somit ein Gesamtsystem entsteht, welches systemübergreifende Transportaufträge durchführen kann. Im Unterschied zu einer rein virtuellen Kopplung auf Basis von Simulationssystemen werden bei diesem Ansatz tatsächlich existierende Förderanlagen gekoppelt. Dies ermöglicht einen praktischen Nachweis der Funktionalität des in diesem Forschungsprojekt entwickelten Kollaborationswerkzeugs. Mit diesem Vorgehen wurde eine virtuelle Inbetriebnahme an realen Anlagenteilen durchgeführt, wie sie zukünftig in heterogenen Großprojekten eingesetzt werden.

Im Folgenden werden zunächst die existierenden Förderanlagen und ihre agentenbasierten Materialflussteuerungen beschrieben. Im Anschluss daran wird ein beispielhafter Anwendungsfall entwickelt und das daraus entstehende Gesamtsystem beschrieben. Diese Tätigkeiten werden bereits durch die KoDeMat-Plattform unterstützt. Ein wichtiger Bestandteil ist die kollaborative Definition der Lastübergabenschnittstelle zwischen den beiden Versuchsanlagen. Diese Schnittstelle wird jeweils in die existierenden dezentralen Anlagensteuerungen integriert.

Zur virtuellen Inbetriebnahme werden beide Anlagensteuerungen über ein VPN-Netzwerk mit der KoDeMat-Plattform gekoppelt. Diese Kopplung besteht zum einen aus einer Schnittstelle der Anlagensteuerungen zum Visualisierungsmodul, so dass über Statusänderungsnachrichten, ein aktuelles Abbild des Gesamtsystemzustandes im laufenden Betrieb dargestellt werden kann. Zum anderen werden Anlagensteuerungen an das Schnittstellenverwaltungsmodul gekoppelt, in dem die Lastübergabeschnittstelle entwickelt wird. Diese Kopplung ermöglicht den Test der Lastübergabe im realen Anlagenbetrieb. Die virtuelle Inbetriebnahme wird durch eine Videoübertragung der realen Versuchsanlagen unterstützt, die eine visuelle Validierung der Funktionalität der KoDeMat-Plattform ermöglicht.



Abbildung 9.1: Versuchsanlage des FLW am Fraunhofer IML

Beschreibung der Versuchsanlage am Lehrstuhl FLW

Für das Kopplungsprojekt wird eine Versuchsanlage am benachbarten Fraunhofer Institut für Materialfluss- und Logistik genutzt (siehe Abbildung 9.1). Diese Stetigförderanlage für Behälter verfügt über 108 separat angetriebene Rollen- und Gurtförderer sowie 18 Weichen. Sie besteht aus einem inneren und einem äußeren Förderkreislauf, 4 Kommissionierbahnhöfen und zwei Lastübergabestellen an ein fahrerloses Transportsystem. Die Anlage wurde mit der Zielsetzung einer dezentralen Steuerung konstruiert und besitzt eine große Anzahl von Lichtschranken und RFID-Lesegeräten, die es ermöglichen Lastobjekte auf fast jedem Förderlement aufzugeben oder abzunehmen. Die agentenbasierte Steuerung kann diese Ereignisse registrieren und entsprechend reagieren. Damit kann jedes der 108 einzeln angetriebenen Fördererelemente als Lastübergabepunkt verwendet werden. Die Anlage kann Behälter mit einer maximalen Geschwindigkeit von 1 m/s befördern. Durch den ausschließlichen Einsatz von Frequenzumrichtern ist die Geschwindigkeit der Antriebe stufenlos regelbar.

Zur Identifizierung der vorbeifahrenden Stückgüter wird RFID-Technologie im UHF-Frequenzbereich (865 MHz) eingesetzt. Insgesamt befinden sich auf der Anlage mehr als 40 Schreib-/Lesestationen, an denen die auf den Gütern angebrachten passiven Transponder identifiziert,

gelesen und beschrieben werden können. Anders als in klassischen Anlagen üblich, wird die Versuchsanlage nicht von einer zentralen SPS gesteuert. Stattdessen sind direkt an der Fördertechnik neun Industrie-PCs (IPC) verbaut. Jeder IPC steuert einen eigenen Teil der Anlage, wobei alle Sensoren und Aktoren eines Förderers gemäß einer strikten Modularisierung stets an den gleichen IPC angeschlossen sind. Als Betriebssystem wird ein Linux mit einem 2.4er Kernel samt Echtzeiterweiterung verwendet. Dies erlaubt es neben den SPS-Programmen beliebige in Hochsprachen (wie Java oder C/C++) entwickelte Software auszuführen. Die verwendete agentenbasierte Anlagensteuerung basiert auf einem in Java geschriebenen und von Lehrstuhl FLW entwickelten Framework.

Beschreibung der Versuchsanlage am Lehrstuhl fml

Kernstück der Versuchsanlage am Lehrstuhl fml ist eine Elektrohängebahnanlage (EHB). Die EHB-Anlage verfügt neben einer Kreisbahn mit drei Weichen über ein Kranfeld, das von einem Einträgerkran befahren wird (siehe Abbildung 9.2). Auf den Schienen befinden sich zwei Katzen zur Aufnahme von VDA-Behältern, die sowohl automatisch wie auch manuell Lager- und Fördererplätze ver- und entsorgen können. Rollenbahnelemente und Übergabeplätze dienen der Demonstration der Verkettungsmöglichkeiten. Transportieren lassen sich VDA-Kleinladungsträger im Bereich bis 50 kg. Für das automatisierte Greifen von VDA-Behältern wurde ein intelligentes Lastaufnahmemittel entwickelt und patentiert, das Positionierungenauigkeiten im Bereich mehrerer Zentimeter ausgleichen kann.

Die Versuchsanlage dient der Erforschung und Entwicklung dezentraler Steuerungskonzepte auf Basis von RFID-Technologie und Agentensystemen für den innerbetrieblichen Materialfluss. Anders als bei heutigen Materialflusssystemen üblich sind sowohl die Katzen als auch die Weichen und Krane autonome Elemente, die selbstständig Entscheidungen treffen und über flexible und erweiterbare Kommunikationsschnittstellen verfügen. Übergeordnete Steuerungsinstanzen wie z.B. Materialflussrechner sind nicht vorhanden. Die verwendete Kommunikations- und Steuerungsarchitektur erlaubt es zudem, virtuelle Fahrzeuge und Streckenstücke in den Anlagenbetrieb zu integrieren. Auf diese Weise wird ein Mischbetrieb von realen und simulierten Modulen



Abbildung 9.2: Versuchsanlage des fml am TU München

ermöglicht, so dass die Folgen geplanter Änderungen bzw. Erweiterungen eines Materialflusssystems im Voraus besser abgeschätzt werden können. Eine topologische Besonderheit der Anlage ist die Fähigkeit die Katzen über den Kran an eine beliebige Position innerhalb des vom Kran abgedeckten Hallenbereichs zu positionieren.

9.1 Modellierung eines virtuellen Gesamtsystems

Im Gegensatz zu einem Projekt zur Planung einer neuen Anlage besteht bei der Entwicklung des Gesamtsystems aus zwei existierenden Versuchsanlagen die Herausforderung ein plausibles Gesamtsystem zu erstellen, welches einen Testbetrieb unterstützt, der genügend nah an dem Verhalten eines realen Fördersystems entspricht. Die beiden Anlagen können im Rahmen des Forschungsprojektes mechanisch nicht verändert werden. Als großer Vorteil sind die unterschiedlichen Fördersysteme der beiden Versuchsanlagen anzusehen. Die Stetigförderanlage auf dem Hallenboden und die aufgeständerte Elektrohängebahnanlage können sehr flexibel übereinander angeordnet werden. Ein weiterer Vorteil ist die bereits genannte freie Positionierbarkeit der EHB-Katzen in Kranbereich sowie die Möglichkeit der Stetigförderanlage Behälter auf beliebigen Förderelementen entgegenzunehmen.

Aufgrund dieser Gegebenheiten sind eine Vielzahl von möglichen Gesamtlayoutvarianten denkbar.

In einem ersten Schritt werden die existierenden statischen Anlagenteile als 3D-Objekte modelliert und der KoDeMat-Visualisierungsplattform als Import zur Verfügung gestellt (siehe Abbildung 9.3). Über die kollaborativen Layoutfunktionen ist anschließend ein kollaboratives Arbeiten am Layout des Gesamtsystems möglich. Die Modellierung der 3D-Objekte erfolgt über CAD-Werkzeuge bzw. durch Import existierender Modelle. Die Erfahrung bei der Entwicklung des Demonstrationsszenarios zeigt, dass auch existierende 3D-Modelle nicht immer direkt verwendet werden können, sondern für die Verwendung in der Kollaborationsplattform angepasst werden müssen. Dies betrifft insbesondere den Import von sehr detaillierten 3D-Modellen aus Konstruktionsprogrammen. Des Weiteren muss auf die geeignete Aufteilung der Submodelle geachtet werden. In der KoDeMat-Plattform können jeweils nur ganze 3D-Modelle als Komponenten importiert und verwendet werden. Einzelne Anlagenteile, die im Rahmen der Anlagensvisualisierung getrennt bewegt und positioniert werden sollen, müssen als separate Objekte in die Plattform importiert werden.

Für die Stetigförderanlage des Lehrstuhls FLW existieren sehr detaillierte 3D-Konstruktionsmodelle des im Datenformat des CAD-Programms Inventor. Diese Modelle können relativ einfach über Konvertierungswerkzeuge in ein Format transformiert werden, das den Import in die KoDeMat-Visualisierung ermöglicht. Der Detailgrad dieser Modelle ist jedoch so hoch, dass im laufenden Betrieb der Visualisierungsplattform beim Laden und Anzeigen dieser Modelle große zeitliche Verzögerungen entstehen, die den Arbeitsablauf stören. Daher wird im Demonstrator mit vereinfachten Modellen gearbeitet, die einen fließenden Ablauf ermöglichen. Diese Vorgehensweise entspricht der Realität im Projektverlauf, da zum Import-Zeitpunkt die genauen 3D-Konstruktionsmodelle üblicherweise nicht vorhanden sind.

Für die Elektrohängebahn des Lehrstuhl fml existieren 3D-Modelle eines bereits bestehenden Visualisierungswerkzeugs. Diese haben den geeigneten Abstraktionsgrad für die KoDeMat-Plattform. Die Modelle der Versuchsanlage sind in folgende Teilmodelle untergliedert: eine

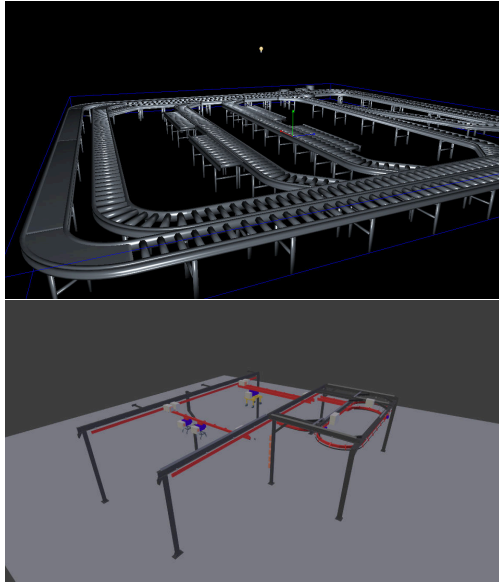


Abbildung 9.3: Existierende statische Anlagenteile modelliert als 3D-Objekte

Einschienenkatze, den Einträgerkran, Einträgerweichen- und schienen, das Lastaufnahmemittel. Aus diesen Teilmodellen lässt sich die Gesamtanlage zusammenstellen. Über geeignete Nachrichten kann so eine Visualisierung des Anlagenzustandes, wie die Position der Katzen und des Krans, separat erfolgen.

Nach der Import der modellierten Teilsysteme in die KoDeMat- Plattform, können diese mit Hilfe der Editorfunktionen zu einem Gesamtsystem zusammengefügt werden (siehe Abbildung 9.4). Diese Arbeiten geschehen kollaborativ durch die Forschungsstellen in München und Dortmund und werden neben der gemeinsamen Arbeit an der Positionierung der Teilmodelle durch die Verwendung der Notizen vereinfacht. Es entsteht ein Gesamtsystem, in welchem die Elektrohängebahnanlage über der Stetigförderanlage so positioniert ist, dass einerseits ihre Ständer an freien Stellen stehen und nicht die unteren Förderwege blockieren und andererseits der EHB-Kranbereich einen großen Teil der Stetigförderanlage abdeckt. Als Lastübergabepunkt

kommen mehrere Förderer in Frage, es wird ein Kommissionierbahnhof gewählt, da so der Förderbetrieb auf den Umlaufstrecken der Stetigförderanlage nicht gestört wird.

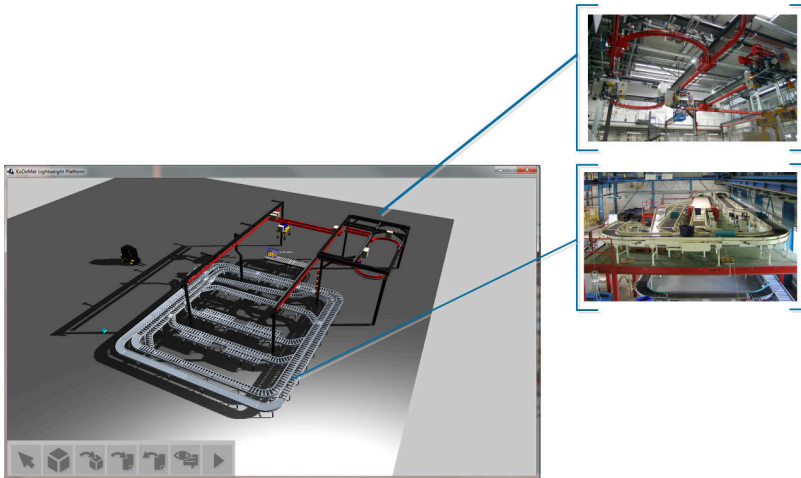


Abbildung 9.4: 3D-Modell eines virtuellen Gesamtsystems. Mithilfe der Editorfunktionen der KoDeMat-Plattform, können die modellierten Teilsysteme zu einem Gesamtsystem zusammengefügt werden

Mit Modellierung und Import der Teilsystemmodelle und der kollaborativen Entwicklung des Layouts kann das virtuelle Gesamtsystem bereits in der Visualisierung betrachtet werden. Weitere Planungsschritte und Aufgaben können mit Hilfe der Kommentarfunktion direkt an die betroffenen Positionen des virtuellen Systems eingetragen werden. Dies erleichtert die weiteren Arbeitsschritte der Anbindung von Materialflusststeuerungen und Simulationswerkzeugen.

9.2 Anbindung der agentenbasierten Steuerungen

Die agentenbasierte Materialflusststeuerung der Stetigförderanlage des FLW basiert auf einer *Web-Service-Architektur*, die es ermöglicht, dass die Agentenprogramme über eine Standard-XML-Schnittstelle kommunizieren können. Die Agentenprogramme laufen in einer Java-Laufzeitumgebung auf unter Linux betriebenen Beckhoff-Industrie-PCs und übernehmen jeweils die Steuerung eines Fördermoduls. Über

Konfigurationsdateien wird die Topologie der Anlage eingelesen und automatisch die Nachbarschaftbeziehungen der Fördereragenten hergestellt, so dass diese über dynamische Routingalgorithmen den Transport von Ladungsträgern vornehmen können. Die Materialflusssteuerung abstrahiert intern von der mechanischen Ausführung der einzelnen Fördererlemente, so dass aus Sicht der Anbindung eine direkte Abbildung zwischen 3D-Modellen und Fördereragenten besteht. Die Kopplung wird über die Einbindung der Hazelcast-Bibliotheken in das Steuerungsprogramm durchgeführt. So entsteht eine sehr einfache und robuste Anbindung an den KoDeMat-Server, über die sowohl der jeweilige Betriebszustand der autonomen Fördererlemente ermöglicht wird, als auch die Kommunikation zwischen den Steuerungen der Versuchsanlagen ermöglicht. Eine Eigenschaft dieses Agentensystems ist der vollkommen dezentrale Ansatz, bei der keine zentrale Datenhaltung der Agenten erfolgt. Jeder Agent verfügt über eine eigene Anbindung an den KoDeMat-Server (siehe Abbildung 9.5).

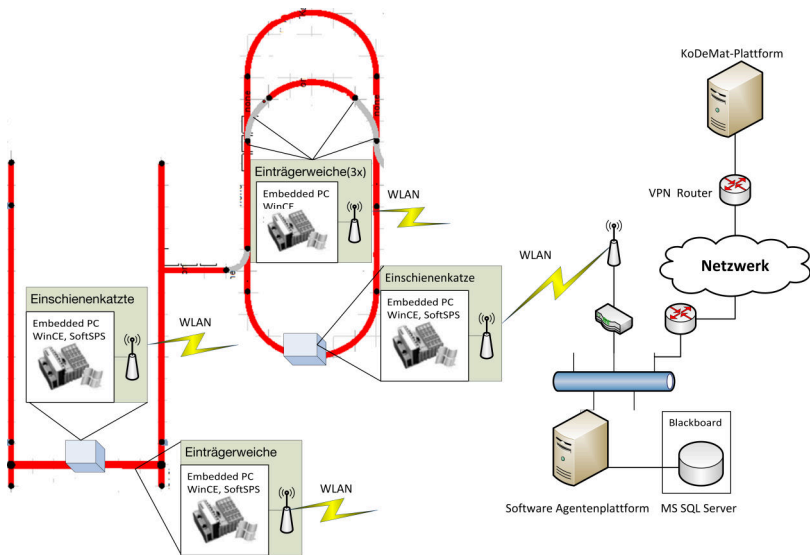


Abbildung 9.5: Netzwerdiagramm der Anbindung einer agentenbasierten Steuerung zur KoDeMat-Plattform

Das dezentrale agentenbasierte Steuerungskonzept der EHB-Anlage

in der Versuchshalle des Lehrstuhls fml wird in [GHM10], [Chi10] beschrieben. Die Steuerungshardware für die Fördermodule besteht aus Industrie-PCs der Firma Beckhoff, die mit Windows CE 5 und einer Soft-SPS ausgestattet sind. Für die Kommunikation der Fördermodule untereinander wird Ethernet bzw. WLAN eingesetzt. Auf den Industrie-PCs läuft die JADE-Agentenplattform, in der die Förderagenten eingebettet sind. Diese Agenten bieten die Dienste der Anlage an, übernehmen den Datenaustausch zwischen den Fördermodulen und die Interaktion mit dem Benutzer. Auf diese Dienste kann über das Service-Directory der JADE-Plattform zugegriffen werden, so dass die Anlage über Standard-PCs bedient werden kann. Ein besonderes Merkmal dieser Agentenarchitektur ist der Einsatz eines Blackboards zur Speicherung systemweit relevanter Daten wie z. B. der Anlagentopologie, Wegreservierungen oder offener Transportaufträge (siehe Abbildung 9.6). Der Rechner mit der Agentenplattform und dem Blackboard wird per TCP-Verbindung mit der KoDeMat-Plattform gekoppelt. Für die Anbindung der Agentenplattform an den KoDeMat-Server wird ein Schnittstellenagent verwendet, der die KoDeMat-Bibliotheken verwendet. Er übernimmt die Kommunikation zwischen den zwei Plattformen. Alle relevante Systemdaten, wie z.B. Transportauftrags- oder Fahrzeugstatus, werden vom Blackboard abgerufen. Auf Seiten der KoDeMat-Plattform werden empfangene TCP-Nachrichten direkt als Hazelcast-Nachrichten weiterverarbeitet. So wird der aktuelle Anlagenzustand, wie er im Blackboard gespeichert ist, als KoDeMat-Visualisierungsereignis gespeichert. Dies wirkt sich direkt auf die Positionen der Teilmodelle in der Visualisierung aus. Bewegt sich beispielsweise eine der EHB-Katzen, so wird dieses Bewegungsereignis im Blackboard gespeichert, von dort per TCP-Verbindung an den KoDeMat-Server gesendet und dort als Positionsänderung der EHB-Katzen-3D-Modelle interpretiert. Diese Positionsänderung wird über die KoDeMat-Architektur an alle Visualisierungsclients weitergesendet.

In Abbildung 9.7 ist die Kopplung der EHB-Versuchsanlage mit der KoDeMat- Visualisierungsumgebung dargestellt. In diesem Testszenario wird der Kran manuell mit der Funksteuerung bedient. Die Fahrzeugagenten melden auch in diesem Handbetrieb die Positionen laufend an das Blackboard. Über den Schnittstellenagent wird der aktuelle Anlagenzustand in die KoDeMat- Visualisierungsumgebung

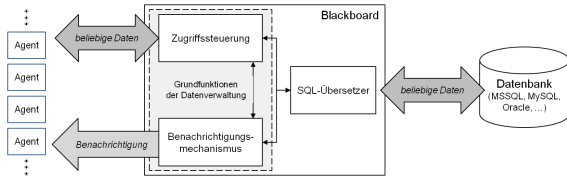


Abbildung 9.6: Funktionen und Komponenten eines Blackboardsystems [Chi10, S. 123].

übertragen. Der Schnittstellenagent ist so programmiert, dass er bei Fehlen der KoDeMat-Plattform keinen kritischen Programmabbruch der Steuerung verursacht. Dies ermöglicht einen unabhängigen Betrieb der Versuchsanlage.

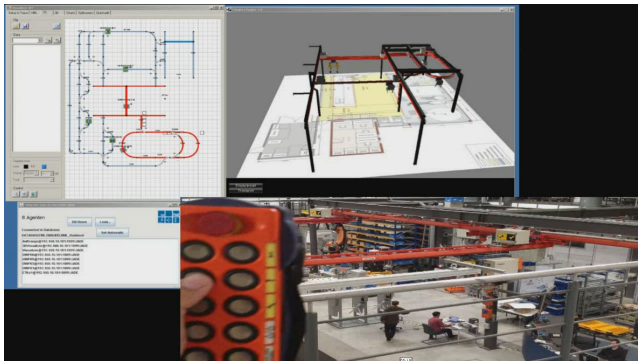


Abbildung 9.7: Kopplung der EHB-Versuchsanlage mit der KoDeMat-Visualisierungsumgebung

Der Datenaustausch zwischen den Anlagensteuerungen wird über Hazelcast-Message-Queues der KoDeMat-Plattform durchgeführt. Die Inhalte der so übertragenen Nachrichten sind so in der Kollaborationsplattform verfügbar und können über die Schnittstellenverwaltung bearbeitet werden.

9.3 Anbindung der Simulationsumgebungen

An beiden Forschungsstellen existieren Simulationswerkzeuge und –modelle, welche die Versuchsanlagen abbilden und virtuell lauffähig machen. Diese Simulationswerkzeuge bilden die physikalischen Eigenschaften der Versuchsanlagen detailliert ab, so dass dort die jeweiligen agentenbasierten Materialflusststeuerungen direkt und ohne Veränderung verwendet werden können. Die Java–basierte Simulationsumgebung der Stetigförderanlage liefert die simulierten Sensorereignisse direkt an das Agentensystem, welches diese im gleichen Nachrichtenformat erhält wie auf der Realanlage. Daher wird auch der simulierte Anlagenzustand über die bereits bestehende Kopplung des Agentensystems mit der KoDeMat–Plattform verwendet.

Die existierende Simulation der Elektrohängebahn liefert den Status und die Position der EHB–Katzen an die JADE–Agentenplattform (siehe Abbildung 9.8). Für die Positionsbestimmung der Fahrzeuge wird zum einen ein Wegmesssystem auf Basis eines Lochstreifens verwendet. Dieser kodiert eine absolute Millimeterposition und wird für die Feinpositionierung beim Erreichen eines Lastübergabeplatzes verwendet. Eine 2D–Visualisierungsumgebung dient zur grafischen und textuellen Anzeige der auf dem Blackboard gespeicherten Daten und wird zu diesem Zweck über alle Datenänderungen vom Blackboard benachrichtigt. Das Simulationswerkzeug verwendet die gleiche Schnittstelle zur Übertragung der simulierten Positionen an das Blackboard. Von dort aus wird die bestehende Kopplung zum KoDeMat–System genutzt.

Das im Folgenden beschriebene TestszENARIO und die kollaborative Schnittstellenentwicklung wird zuerst auf Basis dieser der simulierten Versuchsanlagen durchgeführt. Da diese aus Sicht des KoDeMat–Systems transparent sind, kann anschließend der Inbetriebnahmetest an den Realanlagen ohne weitere Änderung durchgeführt werden.

9.4 Definition der Testläufe

Zur Durchführung der Testläufe wird folgender Ablauf definiert, der einen systemübergreifenden Transportauftrag darstellt, welcher in der Stetigförderanlage beginnt und in der EHB–Anlage endet. Es wird auch der Rücktransport definiert, so dass eine fortlaufende Zirkulation

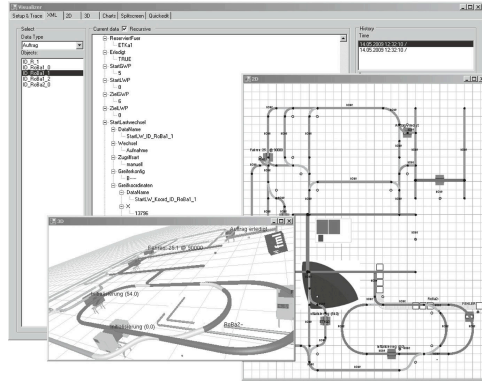


Abbildung 9.8: Screenshot der Visualisierungsumgebung der existierenden Versuchsanlagen mit textueller und grafischer Anzeige ([GHM10, S. 268])

des Testbehälters realisiert ist. Abbildung 9.9 stellt einen Screenshot dieses Prozesses dar.

- Ein VDA-KLT-Behälter wird in Kommissionierbahnhof A auf die Stetigförderanlage gestellt und manuell ein Gesamtsystem-Transportauftrag zur Zielposition in der EHB-Anlage ausgelöst.
- Der Behälter wird zum Übergabepunkt zwischen Stetigförderanlage und EHB-Anlage befördert, der sich in Kommissionierbahnhof B befindet. Ist der Behälter dort angekommen findet das Lastübergabeprotokoll statt, welches kollaborativ in der Schnittstellenverwaltung entwickelt wird. Es wird eine Lastübergabeanfrage vom Förderagenten des Kommissionierbahnhofs an die EHB-Anlage gesendet.
- Ein neuer Fahrauftrag wird generiert, über dessen Zuteilung die EHB-Einschienenkatzen selbständig miteinander verhandeln. Es werden hier z.B. die aktuelle Position zum Übergabepplatz berücksichtigt. Eine der EHB-Katzen nimmt den Auftrag an und berechnet selbständig einen Weg von ihrem eigenen Standort zum Übergabepplatz.

- Bei der Lastübergabe zwischen Rollenförderer und EHB-Katze findet eine Abstimmung zwischen den zwei zuständigen Agenten statt. Insbesondere erteilt der Rollenförderer eine Stapelfreigabe, so dass die EHB-Katze den Behälter sicher aufnehmen oder absetzen kann.
- Die EHB-Katze führt eine neue Wegplanung durch und fördert den aufgenommenen Behälter zu seinem Ziel. Dabei fordert die EHB-Katze evtl. Positionsänderungen von den Weichen oder der Kranbrücke an. Die EHB-Katzen sind zusätzlich in der Lage anderen EHB-Katzen Ausweichaufträge zuzuteilen, wenn diese den Zielort belegt halten oder blockieren.
- Die EHB-Katze transportiert den Behälter zur Zielposition und meldet die erfolgreiche Beendigung des Gesamtsystem-Transportauftrags.

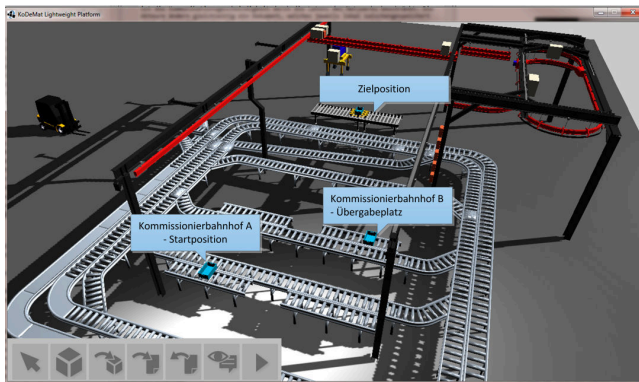


Abbildung 9.9: Definition eines Testlaufs in der KoDeMat-Plattform. Der Transportauftrag beginnt in der Stetigförderanlage und endet in der EHB-Anlage.

Die erfolgreiche Durchführung der hier definierten Testläufe wird durch eine passende Schnittstelle gewährleistet. Die Implementierung eines Gesamtsystem-Transportauftrags erfordert zum einen, dass jede mögliche Zieladresse in beiden System akzeptiert werden muss. Bei Zielen in anderen Teilsystemen kann als Zwischenziel der Lastübergabepunkt

verwendet werden. Dies ist eine generelle Erweiterung, die in die Wegfindungsalgorithmen der Materialflusssteuerungen implementiert werden muss, damit systemübergreifende Transporte durchgeführt werden können. Diese Änderung kann jedoch projektunabhängig erfolgen und Bedarf einer einmaligen Neu- oder Umprogrammierung. Zum anderen erfordert die sichere Lastübergabe die Definition eines Lastübergabeprotokolls am Übergabepunkt. Die grundsätzlichen Funktionen, wie etwa die Fähigkeit zur Gewährleistung der Stapelfreigabe, müssen bereits im Steuerungssystem vorhanden sein. Der folgende Abschnitt beschreibt die kollaborative Schnittstellenentwicklung innerhalb der KoDeMat-Plattform.

9.5 Kollaborative Definition der Auftragstelegramme und der Lastübergabeschnittstelle

Die kollaborative Schnittstellenverwaltung basiert, ähnlich wie die Visualisierungsumgebung, auf der Technik der Distributed Maps. Anstatt die Position von 3D-Modellen abzubilden, werden jedoch Telegrammdefinitionen sowie Telegrammfelddefinitionen verteilt gespeichert. Die Bearbeitung der Telegramme findet in einer klassischen Applikation statt, welche die Telegramme und Telegrammfelder darstellt. Die Benutzer können diese Definitionen kollaborativ bearbeiten und sich bei der Arbeit gegenseitig beobachten. Über diesen Editor werden die Gesamtsystem-Transportaufträge als Telegramm definiert, welches aus einer Liste von Property-Value-Einträgen besteht. Im Fall der Kopplung der beiden Versuchsanlagen wurden alle Telegramme nach der folgenden Struktur definiert:

- ID: Eine eindeutige Kennung des Telegrammfeldes
- Name: Der Name des Telegrammfeldes als Text
- Wert: Der Standardwert des Telegrammfeldes als Text

Des Weiteren wird in jeder Zeile der aktuelle Bearbeiter eines Feldes angezeigt. Sobald ein Benutzer ein Feld bearbeitet, wird dieses für die anderen Benutzer gesperrt. Über einen Codegenerator können die definierten Telegramme im jeweils für die Steuerung passenden Format ausgegeben werden. Die Definition der Codegenerierungsvorlagen muss für jede Materialflusssteuerung einzeln erstellt werden.

Im Fall der Kopplung der beiden handelt es sich um unterschiedliche XML-Dialekte, welche für die Exportvorlagen erstellt wurden. Bei der Definition neuer Telegrammfelder können diese direkt in der passenden Syntax generiert werden. Ein aufwändiges Einlesen der neuen Telegramminhalte entfällt.

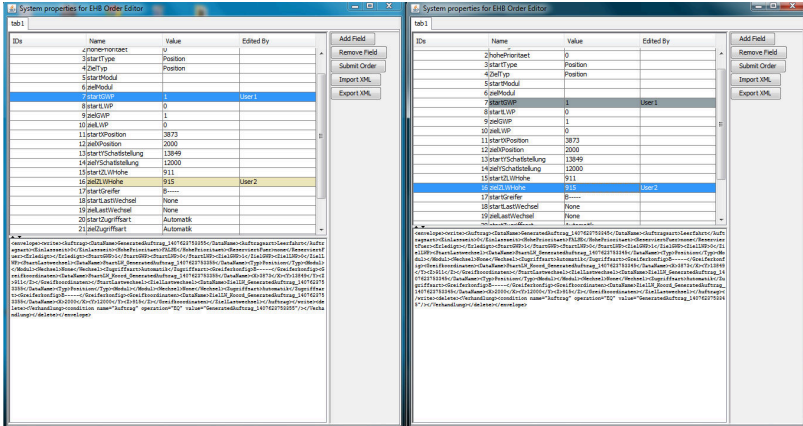


Abbildung 9.10: Kollaborative Bearbeitung des gleichen Telegramm von zwei Benutzern

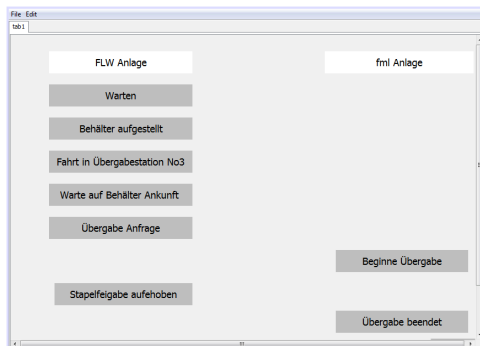


Abbildung 9.11: Programmfenster für die Definition eines Testlauf-Protokolls

Die Abbildung 9.10 zeigt die kollaborative Bearbeitung eines EHB-Transportauftrags, der Teil des Lastübergabeprotokolls ist. Gut zu er-

kennen sind die Telegrammfelddefinitionen sowie die Darstellung des XML-Exports. In der Abbildung sieht man die Darstellung der kollaborativen Bearbeitung des gleichen Telegramm. Zwei Benutzer bearbeiten synchron zwei verschiedene Telegrammfelder. Sie können jeweils anhand der farblichen Markierung sehen, welches Feld der andere Benutzer zur Zeit editiert. Die Änderung an den Werten werden zwischen allen Teilnehmern synchronisiert.

Die Abbildung 9.11 zeigt das Programmfenster, welches das Testlauf-Protokoll darstellt. Von diesem Fenster aus kann durch Mausclick auf die einzelnen Telerammdefinitionen zugegriffen werden.

Nach der vollständigen Definition aller Telegramme kann der Testlauf durchgeführt werden. Dazu können die Telegramme einzeln direkt aus der Kollaborationsumgebung heraus an die Agenten versendet werden. Der vollständige Testablauf wird mit Hilfe der Simulationsumgebungen durchgeführt. Erst bei korrektem Ablauf kann der im folgenden Abschnitt beschriebene verteilte Inbetriebnahmetest des Gesamtsystems erfolgen.

9.6 Verteilter Inbetriebnahmetest des Gesamtsystems

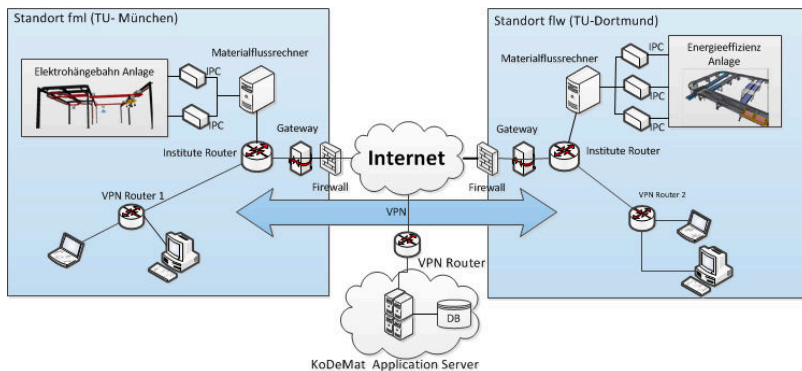


Abbildung 9.12: Netzwerkdigramm der Kopplung zweier Versuchsanlagen mittels der KoDeMat-Plattform

Zur verteilten Inbetriebnahme des Gesamtsystems werden die Simulatoranbindungen durch die realen Versuchsanlagen ersetzt. Dies er-

fordert die Installation der erweiterten KoDeMat-fähigen Steuerung auf den Steuerungsrechnern der Versuchsanlage und die Anbindung an das KoDeMat-VPN. Diese Anbindung erfolgt über portable Lancom Hardware-VPN-Router, die über einen Master-Router ein lokales Netzwerk zwischen den Steuerungen der Versuchsanlagen herstellen. In dieses Netzwerk ist der KoDeMat-Server integriert. Die Abbildung 9.12 stellt dieses Konzept vor. Die Durchführung der Tests erfordert die Bereitstellung von Behältern an den realen Versuchsanlagen, mit denen der Testablauf durchgeführt wird. Die Lastübergabe zwischen den realen Versuchsanlage wird durch Mitarbeiter dargestellt welche zum richtigen Zeitpunkt die Behälter aufnehmen oder abgeben. Die EHB-Katze nimmt den Behälter in der Münchener Versuchsanlage erst dann auf, wenn die Dortmunder Anlage die Stapelfreigabe gesendet hat. Dort wird der entsprechende Behälter manuell von der Fördertechnik heruntergenommen, wenn die EHB-Katze den dortigen Testbehälter in München erfolgreich aufgenommen hat. Zur Überwachung der korrekten Durchführung der verteilten Inbetriebnahme werden Videoübertragungen der jeweils anderen Versuchsanlage auf einen Monitor übertragen. So kann die zeitlich korrekte Lastübergabe überwacht sowie geprüft werden, ob die KoDeMat-Visualisierung den Zustand beider Teilsysteme korrekt wiedergibt.

9.7 Fazit

Mit der Kopplung der realen agentengesteuerten Versuchsanlagen wird die Funktionalität der KoDeMat-Plattform nachgewiesen. Durch den Einsatz der Software in realen Fördersystemen kann so der Einsatz und die Notwendigkeit der KoDeMat-Software demonstriert werden. Die unterschiedlichen architektonischen Ansätze der beiden Materialflusssteuerungen stellen ein gutes Beispiel für die Integration heterogener Teilsystem dar und zeigen die notwendigen geringfügigen Änderungen an diesen Systemen auf, damit diese innerhalb der KoDeMat-Plattform arbeiten können.

Abbildungsverzeichnis

1.1	Vergleich der projektspezifischen Aufwände für Planung, Realisierung und Test bei Materialflussanlagen mit zentraler Steuerung (l.) und dezentral gesteuerten Anlagen (r.) ([Kuz10], S.9)	3
1.2	Beispielhaftes heterogenes Großprojekt	4
2.1	Entwicklungsschritte hin zu einer Digitalen Fabrik im zeitlichen Überblick (vgl. [Sch08])	12
2.2	Tätigkeiten von der Planung bis zum Betrieb einer Materialflussanlage und die wichtigsten Meilensteine (Quelle: Forschungsprojekt <i>Internet der Dinge</i>) ([GHM10, S.175])	22
2.3	Erwartete Veränderung des Zeitaufwands in der Planungsphase (qualitativ) ([Kuz10, S. 153])	24
2.4	Erwartete Veränderung des Zeitaufwands in der Realisierungsphase (qualitativ) ([Kuz10, S. 155])	26
2.5	Erwartete Veränderung des Zeitaufwands in der Inbetriebnahme- /Hochlaufphase (qualitativ) ([Kuz10, S. 157])	28
3.1	Kollaborationsplattform als zentrales Element eines herstellerübergreifenden Engineeringsprozesses	32
3.2	Methodische Aufteilung des Forschungsprojektes in Arbeitspakete	36
3.3	Aufbau der Arbeit	37
4.1	Tätigkeiten bei der Entstehung einer Materialflussanlage	40
4.2	Planungsprozess eines dezentralen Materialflusssystems mit der KoDeMat-Plattform	42
4.3	Hierarchische Struktur eines Materialflusssystems	44
5.1	Vorgehensweise für die Gestaltung der KoDeMat-Plattform	47
5.2	Akteure, die bei einem intralogistischen Projekt zusammenarbeiten	49
5.3	Referenzszenario der KoDeMat-Plattform für ein heterogenes Großprojekt	50
6.1	Softwarearchitektur der KoDeMat-Plattform	58
6.2	Konzept der Datenverteilung und Versionsverwaltung der KoDeMat-Plattform	60
6.3	Cluster des In-Memory-Datenbanksystems von Hazelcast (vgl. [Vee14])	62
6.4	Datenmodell der KoDeMat-Plattform	63

6.5	Beispiel einer synchronen Bearbeitung von 3D-Objekten von zwei Benutzern (User_1 und User_2) (a) User_1 verschiebt das 3D-Objekt eines Fördersystem zu Position B (b) User_2 dreht das Objekt nach rechts. (c) Der User_1 verschiebt das Objekt zu Position C. . . .	66
6.6	Änderungshistorien des 3D-Objektes für das Szenario der Abbildung 6.5. Im Visualisierungswerkzeug werden Zeitstempel, Änderungsinformation und Benutzername für jede neue logische Einheit erweitert.	66
6.7	(l.): serviceorientierte Architektur (engl. SOA), (r.): ereignisgesteuerte Architektur (engl. EDA) [Mar06b] . .	67
6.8	Ereignisgesteuerte Architektur der KoDeMat-Plattform	68
6.9	Lebenszyklus eines Datencontainers	69
6.10	Kommunikationsdiagramm zwischen Softwarekomponenten bei einer Positionsänderung eines 3D-Objektes . .	70
6.11	(a) Definition einer Anlagentopologie im Planungswerkzeug, (b) Die im GEXF-Format exportierte Topologie kann in einer externen Anwendung wie Gephi dargestellt und bearbeitet werden, (c) Der resultierende Graph kann als Datenmatrix visualisiert werden	72
7.1	Positionierung eines 3D-Objekts in der virtuellen Umgebung des KoDeMat-Plattform	76
7.2	Screenshots von Funktionen des 3D- Visualisierungswerkzeugs	77
7.3	Darstellungsmethoden für die Abbildung des Materialflusses (vgl. [AF09, S. 348], [Kam13, S. 13])	79
7.4	Schema eines gerichteten Graphen	80
7.5	Darstellung eines logistischen Netzwerks mittels eines hierarchischen Graphen	81
7.6	Definition von Verbindungen zwischen Systemkomponenten	82
7.7	Visualisierung von agenten-generierten Ereignissen (vgl. [Chi10, S. 122])	83
7.8	Netzwerkarchitektur zur Visualisierung realer Systeme	84
7.9	Einsatzszenario der KoDeMat-Plattform als Visualisierungswerkzeug für Flurförderzeuge	86
7.10	Visualisierung eines Flurförderzeugs mittels der KoDeMat-Plattform	86

7.11	Die 3D-Visualisierungsumgebung von KoDeMat-Plattform wurde erweitert um VR-Brille von Oculus [Ocu14] zu unterstützen	88
7.12	Einsatz eines Multi-Touch-Screens als Interaktionsgerät für das Visualisierungswerkzeug	89
8.1	Sequenzdiagramm des FIPA Request Interaction Protocol [FIPb]	96
8.2	Kommunikation zwischen zwei agentenorientierten Materialflusssystemen mittels Schnittstellenagenten (vgl. [GHM10])	97
8.3	Screenshot des Echtzeiteditors für Nachrichtendefinitionen in der KoDeMat-Plattform. Mehrere Benutzer können gleichzeitig die Schnittstellenspezifikationen synchron definieren.	98
8.4	Dynamische Generierung einer XML-kodierten Agentenkonfigurationsdatei: Die Felder können gleichzeitig von verschiedenen Akteuren bearbeitet werden. Die resultierende XML-Datei kann sofort als Text visualisiert, überprüft und an den Agenten verteilt werden.	99
9.1	Versuchsanlage des FLW am Fraunhofer IML	102
9.2	Versuchsanlage des fml am TU München	104
9.3	Existierende statische Anlagenteile modelliert als 3D-Objekte	106
9.4	3D-Modell eines virtuellen Gesamtsystems. Mithilfe der Editorfunktionen der KoDeMat-Plattform, können die modellierten Teilsysteme zu einem Gesamtsystem zusammengefügt werden	107
9.5	Netzwerkdiagramm der Anbindung einer agentenbasierten Steuerung zur KoDeMat-Plattform	108
9.6	Funktionen und Komponenten eines Blackboardssystems [Chi10, S. 123].	110
9.7	Kopplung der EHB-Versuchsanlage mit der KoDeMat-Visualisierungsumgebung	110
9.8	Screenshot der Visualisierungsumgebung der existierenden Versuchsanlagen mit textueller und grafischer Anzeige ([GHM10, S. 268])	112
9.9	Definition eines Testlaufs in der KoDeMat-Plattform. Der Transportauftrag beginnt in der Stetigförderanlage und endet in der EHB-Anlage.	113

9.10	Kollaborative Bearbeitung des gleichen Telegramm von zwei Benutzern	115
9.11	Programmfenster für die Definition eines Testlauf-Protokolls	115
9.12	Netzwerkdigramm der Kopplung zweier Versuchsanlagen mittels der KoDeMat-Plattform	116

Tabellenverzeichnis

1	In-memory Datenbanksysteme Lösungen (vgl. [Pri13])	61
2	Beispiel eines Composite-Event	69

Abkürzungsverzeichnis

- ACL** Agent Communication Language. 78
- AP** Anschlusspunkt. 67
- API** Application Programming Interface. 70
- CAD** Computer Aided Design. 13
- CIM** Computer Integrated Manufacturing. 13
- CPS** Cyber-Physischen Systeme. 7, 15
- EHB** Elektrohängebahn. 38, 84
- GEXF** Graph Exchange XML Format. 60
- KMU** Kleine und mittlere Unternehmen. 11
- MDD** Model-Driven Development. 20
- UDP** User Datagram Protocol. 71
- VPN** Virtual Private Network. 43
- VR** Virtual Reality. 73
- XML** Extensible Markup Language. 60

Literatur

- [AF09] ARNOLD, D. ; FURMANS, K.: Materialfluss in Logistiksystemen. In: *Materialfluss in Logistiksystemen* (2009)
- [ALK⁺02] ALTINEL, M. ; LUO, Q. ; KRISHNAMURTHY, S. ; MOHAN, C. ; PIRAHESH, H. ; LINDSAY, B. ; WOO, H. ; BROWN, L.: Dbcache: Database caching for web application servers. In: *Proceedings of the 2002 ACM SIGMOD international conference on Management of data* ACM, 2002, S. 612–612
- [Bae05] BAERISCH, S.: Versionskontrollsysteme in der Softwareentwicklung: IZ-Arbeitsbericht Nr. 36. (2005)
- [BHG09] BASTIAN, M. ; HEYMANN, S. ; JACOMY, M.: Gephi: An Open Source Software for Exploring and Manipulating Networks. In: *ICWSM 8* (2009), S. 361–362
- [Bra06] BRAY, J. T. P. T. Paoli: *Extensible Markup Language (XML) 1.0 (Fourth Edition)*. 2006
- [BSWW09] BULLINGER, HJ. ; SPATH, D. ; WARNECKE, HJ. ; WESTKÄMPER, E.: *Handbuch Unternehmensorganisation Strategien, Planung, Umsetzung*. 2009
- [Chi10] CHISU, R.: *Kommunikations- und Steuerungsstrategien für das Internet der Dinge*. München : TUM, Fml, 2010
- [CM07] CHOPRA, S. (Hrsg.) ; MEINDL, P. (Hrsg.): *Supply Chain Management*. 2007
- [Co.11] Co., SoftGuide GmbH : *RPlan – Projektmanagementsoftware*. http://www.softguide.de/prog_i/pi_0594.htm. Version: 2011
- [CTG96] *Kapitel Madefast - Collaborative Engineering over the Internet*. In: CUTKOSKY, M. R. ; TENENBAUM, J. M. ; GLICKSMAN, J.: *Communications of the ACM*. 1996, S. 78–87
- [DR93] DEWAN, P. ; RIEDL, J.: Toward Computer-Supported Concurrent Software Engineering. In: *IEEE Computer*, 26, S. 17-27. (1993)

- [EKTW12] EBNER, A. ; KAMMERGRUBER, F. ; T., Horenburg ; W., Günthner: Logistics and Layout Planning of Construction Equipment on a VR-Multi-Touch-Tablet. In: *12th International Conference on Construction Application of Virtual Reality* (2012), November
- [Emu14] EMULATE3D: *Demo3D, Homepage*. <http://www.demo3d.com/>. Version: 2014
- [Erl09] ERL, T.: *SOA design patterns*. 1st ed. Upper Saddle River, NJ : Prentice Hall, 2009 (Prentice Hall service-oriented computing series from Thomas Erl)
- [FIPa] FIPA: *FIPA ACL Message Structure Specification*. <http://www.fipa.org/specs/fipa00001/index.html>
- [FIPb] FIPA: *FIPA Request Interaction Protocol Specification*. <http://www.fipa.org/specs/fipa00026/index.html>
- [FIPc] FIPA - FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS: *FIPA Abstract Architecture Specification*. <http://www.fipa.org/specs/fipa00001/index.html>
- [FL05] FUH, J. Y. H. (Hrsg.) ; LI, W. D. (Hrsg.): *Advances in collaborative CAD: The-state-of-the-art*. 2005
- [FM09] FURMANS, K. ; MAYER, S.: Vollständig dezentraler und autonomer Flexförderer – Neun Module im Versuchsbetrieb. In: *Hebezeuge Fördermittel* 49 (2009), S. 308–309
- [For14] FORSCHUNGSUNION - WIRTSCHAFT UND WISSENSCHAFT: *Wirtschaft und Wissenschaft begleiten die Hightech-Strategie*. <http://www.forschungsunion.de/>. Version: 2014
- [Gad12] GADATSCH, A.: *Grundkurs Geschäftsprozess-Management: Methoden und Werkzeuge für die IT-Praxis: eine Einführung für Studenten und Praktiker*. Bd. 7. Wiesbaden : Springer Vieweg, 2012
- [Gam04] GAMMA, E.: *Design patterns: Elements of reusable object-oriented software*. 29th printing. Boston : Addison-Wesley, 2004 (Addison Wesley professional computing series)

- [GCK08] GÜNTNER, W. A. ; CHISU, R. ; KUZMANY, F.: Internet der Dinge – Zukunftstechnologie mit Kostenvorteil. In: *f+h Fördern und Heben* (2008), S. 556–558
- [Gen11] GENTLEWARE: *Poseidon for UML*. <http://www.gentleware.de>. Version: 2011
- [GHM10] GÜNTNER, W. A. ; HOMPEL M. ten: *Internet der Dinge in der Intralogistik*. 2010
- [GHW06] GÜNTNER, W. A. ; HEINECKER, M. ; WILKE, M.: *abschlussbericht AiF-FV-Nr. 14021 N „Modulare Materialflusssysteme für wandelbare Fabrikstrukturen“*. 2006
- [GR00] GÜNTNER, W. A. ; REINHART, G.: *abschlussbericht MATVAR –Materialflusssysteme für variable Fertigungssegmente im dynamischen Produktionsumfeld*. 2000
- [GW04] GÜNTNER, W. A. ; WILKE, M.: M2 – Wandelbare Logistikstrukturen in Minifabriken. In: *Arbeits- und Ergebnisbericht des Sonderforschungsbereichs 582: Marktnahe Produktion individualisierter Produkte* (2004)
- [Gü10] GÜNTNER, W. A.: *Planung technischer Logistiksysteme, Vorlesungsskriptum, Lehrstuhl für Fördertechnik Materialfluss Logistik, TUM*. 2010
- [Hau08] HAUSSTÄDTLER, Uwe: *Der Einsatz von Virtual Reality in der Praxis: Handbuch für Studenten und Ingenieure*. Aachen : Shaker Media, 2008
- [Hay09] HAYES, J.: *Collaboration Solution for Engineers:: Engineering Collaboration – Why a White Paper in 2009*. <http://collaboration.engineering.com/pdfs/>. Version: 2009
- [HB13] HEYMANN, S. ; BASTIAN, M. et. a.: *GEXF File Format*. <http://gexf.net/format/>. Version: 2013
- [HG12] HOHENSTEIN, F. ; GÜNTNER, W. A.: Anforderungen und Fähigkeiten gegenwärtiger Stapler-Lokalisierung. In: *Tagungsband: 19. Juni 2012, Helmut Schmidt Universität, Universität der Bundeswehr Hamburg* (2012). ISBN 3868180370

- [HLR11] HOMPEL, M. ten ; LIBERT, S. ; ROIDL, M.: Abschlussbericht AiF-FV-Nr. 15313 N “Erarbeitung von Methoden und Regeln zur Gestaltung agentengestützter, dezentraler Steuerungen für den Einsatz in Komplexen Materialflusssystemen” / FLW. 2011. – Forschungsbericht
- [Hom06] HOMPEL, M. ten: Zellulare Fördertechnik. In: *Logistics Journal* (2006)
- [HRH06] HILDENBRAND, T. ; ROTHLAUF, F. ; HEINZL, A.: Ansätze zur kollaborativen Softwareerstellung. In: *Wirtschafts Informatik* 49 (2006)
- [HSL04] HOMPEL, M. ten ; SONDHOF, U. ; LIBERT, S.: Vorzüge dezentraler autonomer Steuerungssysteme. In: *f+h* (2004)
- [HW07] HÜLSMANN, M. ; WINDT, K.: *Understanding Autonomous Cooperation and Control in Logistics – The Impact of Autonomy, Management, Information, Communication and Material Flow*. Springer, 2007
- [IBM11] IBM: *Rational Software Development Tools*. <http://www.ibm.com/software/rational/>. Version: 2011
- [iFA11] iFAKT GMBH: *Collaborative Engineering bei der iFAKT*. <http://www.ifakt.de/de/kompetenzen/collaborative-engineering>. Version: 2011
- [JHG13] JUNG, M. ; HOHENSTEIN, F. ; GÜNTNER, W. A.: Staplerauge– a framework for camera-based sensor functions on forklift trucks. In: *Proceedings of the International Logistics Science Conference (ILSC)* (2013)
- [Kam13] KAMMERGRUBER, F.: *Mobiles Virtual-Reality-System zur Planung und Schulung in der Logistik*. Garching b. München, Diss., 2013
- [Ker05] KERN, E.M.: *Verteilte Produktentwicklung - Rahmenkonzept und Vorgehensweise zur organisatorischen Gestaltung*. 2005

- [KK02] KERSTEN, W. ; KERN, E.-M.: A concurrent engineering-oriented design Database Representation Model. Computer-Aided Design, Internetunterstützte Zusammenarbeit im Produktentstehungsprozess; In: *VDI-Z 144 Heft 7/8, S.47-49* (2002)
- [KN08] KAMRANI, A. K. (Hrsg.) ; NASR, E. M. (Hrsg.): *Collaborative Engineering: Theory and Practice*. 2008
- [Kra90] KRALLMANN, H.: *CIM: Expertenwissen für die Praxis*. München : Oldenbourg, 1990
- [Kus13] KUSTERER, R.: *JMonkeyEngine 3.0 beginner's guide: Develop professional 3D games for desktop, web, and mobile, all in the familiar Java programming language*. Birmingham : Packt Pub., 2013
- [Kuz10] KUZMANY, F. A.: *Konzeption und Entwicklung von Modulen für das Internet der Dinge*. Garching, Lehrstuhl für Fördertechnik, Materialfluss, Logistik, Techn. Univ. München, Diss., 2010
- [KWH12] KAGERMANN, H. ; WAHLSTER, W. ; HELBIG, J.: *Deutschlands Zukunft als Produktionsstandort sichern - Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0: Abschlussbericht des Arbeitskreises Industrie 4.0*. Berlin : Eigenverlag, 2012
- [LNF13] LAHIRI, T. ; NEIMAT, M.-A. ; FOLKMAN, S.: Oracle TimesTen: An In-Memory Database for Enterprise Applications. In: *IEEE Data Eng. Bull.* 36 (2013), Nr. 2, S. 6–13
- [LR93] LOUGHER, R. ; RODDEN, T.: Supporting Long-term Collaboration in Software Maintenance. In: *Proceedings of the Conference on Organizational Computing Systems* (1993)
- [LRR⁺13] LINDSTRÖM, J. ; RAATIKKA, V. ; RUUTH, J. ; SOINI, P. ; VAKKILA, K.: IBM solidDB: In-Memory Database Optimized for Extreme Speed and Availability. In: *IEEE Data Eng. Bull.* 36 (2013), Nr. 2, S. 14–20

- [Mar06a] MARCZINSKI, G.: Zur Bedeutung der Digitalen Fabrik: Ergebnisse einer aktuellen Umfrage. In: *Zeitschrift für wirtschaftlichen Fabrikbetrieb 101* (2006), Nr. 4, S. 218–221
- [Mar06b] MARECHAUX, J.-L.: *Combining Service-Oriented Architecture and Event-Driven Architecture using an Enterprise Service Bus*. <http://www.ibm.com/developerworks/library/ws-soa-eda-es>. Version: 2006
- [MDTR93] MASHAYEKHI, Vahid ; DRAKE, Janet M. ; TSAI, Wei-Tek ; RIEDL, John: *Distributed, Collaborative Software Inspection*. In: *IEEE Software*, 10. 1993
- [Meb08] MEBES, P.: *Graphen-basierte Modellierungsmethode zur Materialflussverfolgung*, Eidgenössischen Technischen Hochschule Zürich, Diss., 2008
- [MFR94] MASHAYEKHI, V. ; FEULNER, C. ; RIEDL, J.: CAIS: Collaborative Asynchronous Inspection of Software. In: *Proceedings of the 2nd ACM SIGSOFT Symposium on Foundations of Software Engineering* (1994)
- [MS02] MONPLAISIR, L. ; SALHIEH, S.M.: *Collaborative product design and development*. 2002
- [Ocu14] OCULUS, VR: *Oculus Rift*. [\url{http://www.oculus.com/}](http://www.oculus.com/). Version: 2014
- [OMG13] OMG: *Object Management Group, Homepage*. <http://www.omg.org/>. Version: 2013
- [OVFK10] OVERMEYER, L. ; VENTZ, K. ; FALKENBERG, S. ; KRÜHN, T.: Interfaced multidirectional small scaled modules for interlogistics operations. In: *Logistics Research 2* (2010)
- [Pri13] PRINS, R.: *In-Memory Data Grids*. <http://architects.dzone.com/articles/memory-data-grids>. Version: March 2013
- [RBGH06] RASHID, A. ; BEHM, A. ; GEISSER, M. ; HILDENBRAND, T.: *Kollaborative Softwareentwicklung - Zum Kollaborationsbegriff*. 2006

- [Ree01] REESE, A. K.: *The Collaboration Question. Supply and Demand Chain Executive*. 2001
- [Rus13] RUSSWURM, S.: Software: Die Zukunft der Industrie. In: SENDLER, U. (Hrsg.): *Industrie 4.0 – Beherrschung der industriellen Komplexität mit SysLM*. Berlin : Springer Vieweg, 2013, S. 21–36
- [SBLD98] SATTES, I. ; BRODBECK, H. ; LANG, H. ; DOMEISEN, H.: *Erfolg in kleinen und mittleren Unternehmen: Ein Leitfaden für die Führung und Organisation in KMU*. 2. Auflage. Zürich : Vdf Hochschulverlag an der ETH, 1998
- [Sch90] SCHEER, A.-W.: *CIM-Strategie als Teil der Unternehmensstrategie*. Berlin : Springer, 1990
- [Sch99] SCHÖTTNER, J.: *Produktdatenmanagement in der Fertigungsindustrie: Prinzip - Konzepte - Strategien*. München : Hanser, 1999
- [Sch08] SCHACK, R.: *Methodik zur bewertungsorientierten Skalierung der Digitalen Fabrik*. München : Herbert Utz Verlag, 2008
- [Sen13] SENDLER, U.: *Industrie 4.0– Beherrschung der industriellen Komplexität mit SysLM (Systems Lifecycle Management)*. Berlin : Springer, 2013
- [SGG⁺13] SPATH, D. ; GANSCHAR, O. ; GERLACH, S. ; HÄMMERLE, M. ; KRAUSE, T. ; SCHLUND, S.: *Produktionsarbeit der Zukunft - Industrie 4.0: Studie*. Stuttgart : Fraunhofer Verlag, 2013
- [Sie14] SIEMENS, AG: *Plant Simulation, Homepage*. <http://www.plm.automation.siemens.com>. Version: 2014
- [SK97] SPUR, G. (Hrsg.) ; KRAUSE, F.-L. (Hrsg.): *Das virtuelle Produkt: Management der CAD-Technik* Hanser Verlag München. 1997
- [Spi12] SPILLNER, A.: *Entwicklung, Stand und Perspektiven der digitalen Fabrik*. Shaker, 2012

- [SRHM97] STEIN, M. ; RIEDL, J. ; HARNER, S. J. ; MASHAYEKHI, V.: A Case Study of Distributed, Asynchronous Software Inspection. In: *Proceedings of the 19th International Conference on Software Engineering* (1997)
- [Tjo11] TJORA, A.: *Use Case Specification Methodology: Deliverable 2.1 for Finest EU Projekt.* [\url{http://www.finest-ppp.eu/index.option=com\textunderscorecontent\&view=article\&id=3\&Itemid=108.html}](http://www.finest-ppp.eu/index.option=com\textunderscorecontent\&view=article\&id=3\&Itemid=108.html). Version: 2011
- [USD11] U., Bracht ; S., Wenzel ; D., Geckler: *Digitale Fabrik: Methoden und Praxisbeispiele*. Berlin : Springer, 2011
- [VDI08] VDI-GESELLSCHAFT PRODUKTION UND LOGISTIK: *VDI-Richtlinie 4499; Digitale Fabrik - Grundlagen*. 2008
- [Vee14] VEENTJER, P.: *Mastering Hazelcast: The Ultimate Hazelcast Book*. 2014
- [Vet03] VETTER, E.: *Softwaresimulation von Laserscannern in einer virtuellen 3D-Umgebung*, Diss., 01.01.2003
- [WBK03] WESTKÄMPER, E. ; BIRSCHENK, S. ; KUHLMANN, T.: Digitale Fabrik – nur was für die Großen? In: *wt Werkstattstechnik online 93* (2003), Nr. 1-2, S. 22–26
- [Wen04] WENZEL, S.: Die Digitale Fabrik: Ein Konzept für interoperable Modellnutzung. In: *Industrie Management 20* (2004), Nr. 3, S. 54–58
- [Wes01] WESTKÄMPER, E.: Visionen für die Digitale Fabrik. In: *Zeitschrift für wirtschaftlichen Fabrikbetrieb 96* (2001), Nr. 3, S. 96–100
- [Wie01] WIENDAHL, H-P.: Auf dem Weg zur Digitalen Fabrik. In: *Werkstattstechnik online 92* (2001), Nr. 4, S. 121
- [Wir11] WIRTSCHAFTSLEXIKON, Gabler: *Collaborative Engineering.* <http://wirtschaftslexikon.gabler.de/Definition/collaborative-engineering.html>. Version: 2011

- [WSX⁺02] WANG, L. H. ; SHEN, W. M. ; XIE, H. ; NEELAMKAVIL, J. ; PARDASANI, A.: *A concurrent engineering-oriented design Database Representation Model. Computer-Aided Design Collaborative conceptual design state of the art and future trends.* 2002