

1. Einleitung

„Wege entstehen dadurch, dass man sie geht.“

Franz Kafka

Die rechnergestützte Automatisierung ermöglicht, Prozesse in der Produktion und im Handel effizienter zu gestalten, und trägt somit zur Verbesserung der Lebensqualität bei. Damit steigen jedoch die Anforderungen an die automatisierten Systeme insgesamt, weil sich gleichzeitig die Geschäftstätigkeiten der Unternehmen verstärken und sich das Lebenstempo der modernen Gesellschaft erhöht (vgl. [MuJ00, S. 13]). Daraus ergibt sich eine immer weiter steigende Aufgabenkomplexität, die mithilfe von Computern bewältigt werden muss.

Die in dieser Arbeit betrachteten Automatisierungssysteme sind Materialflusssysteme der Produktion und Logistik, die in ihrer Größe und Komplexität einer stetigen Weiterentwicklung unterliegen (vgl. [SRBF⁺08, S. 110], [tH03]). Zurzeit werden insbesondere die Individualisierung der Produktion und die Atomisierung der Warenströme als wichtigste Faktoren gesehen, welche die Komplexitätssteigerung hervorrufen (vgl. [tH10]). Von Materialflusssystemen wird dabei, neben Erfüllung der logistischen Leistung, auch Wandelbarkeit – das heißt Flexibilität und Anpassungsfähigkeit – erwartet (vgl. [GCK08b], [Koc06], [GüW02], [Bun05]). Eine entscheidende Rolle bei dieser Aufgabe spielen Materialflussteuerungen, die zu den größten und komplexesten Softwareapplikationen zählen [tH08, S. 322]. Die Intention der vorliegenden Arbeit ist in diesem Zusammenhang, einen Beitrag zur Reduktion des Entwicklungsaufwands solcher Systeme zu leisten.

1.1. Problemstellung

Aktuelle Trends begegnen der Forderung nach Flexibilität mit hoch modularen Systemkonzepten (vgl. [Kuz10], [BSM⁺09], [Wil06]). Fortschritte in der Mikroelektronik, Kommuni-

kationstechnik und Informationstechnologie ermöglichen dabei neue Lösungen und Applikationen in der Prozessautomatisierung (vgl. [MaF09], [Bul07], [FlM05]). Ein Wildwuchs solcher Lösungen in den letzten Jahren erschwert die herstellerübergreifende Projektarbeit und erfordert neue Standards und Richtlinien [Irr07].

In der Steuerungstechnik sind diese Trends dadurch gekennzeichnet, dass die Informationstechnologie immer mehr in Bereiche der prozessnahen Automatisierung eindringt. Um die Komplexität zu beherrschen, werden hier immer häufiger höhere Programmiersprachen und objektorientierte Entwicklungsmethoden verwendet. In den letzten Jahren wird der Einsatz von Softwareagenten zur Steuerung technischer Systeme und Anlagen besonders intensiv erforscht (vgl. [Mön06], [Bra00], [BJW04], [KIK03]). Einerseits verspricht der Gebrauch solcher Werkzeuge viele Vorteile. Darunter zählen beispielsweise bessere Anpassungsfähigkeit und Robustheit entstehender Softwaresysteme (vgl. [Nop10a, S. 229], [Ris02, S. 213]). Andererseits lassen sich diese Vorteile nur nutzen, wenn die Werkzeuge richtig und effizient eingesetzt werden. Mit anderen Worten müssen diese Werkzeuge in einem neuen Umfeld der Anlagensteuerung beherrschbar sein. Erst dann kann der Entwicklungsprozess vereinfacht und können die Entwicklungskosten reduziert werden.

Die Multiagentensysteme bzw. die Agententechnik sind universale Ansätze zur Reduktion der Systemkomplexität (vgl. [Jen00]). Im Bereich der Automatisierung und Steuerung existieren bereits erfolgreiche Realisierungen, die auf dieser Technik aufbauen (vgl. [Koh08], [Gre08], [SuB01]). Jedoch handelt es sich dabei um proprietäre Einzellösungen, die untereinander nicht kompatibel sind und sich kaum auf andere Probleme bzw. Anlagen übertragen lassen.

Ein Hindernis bei der Integration von Multiagentensystemen in bestehende IT-Strukturen stellt dabei die Existenz einer Vielzahl von Agentensprachen, Systemarchitekturen und Laufzeitumgebungen dar [Ris02, S. 212]. Die gewünschte Vereinheitlichung verspricht der FIPA-Standard der Foundation for Intelligent Physical Agents (vgl. [Fou09b]). Die unabhängig, aber unter Verwendung von FIPA-Spezifikationen entwickelten Systeme lassen sich mit relativ geringem Integrationsaufwand zusammenführen und mit IT-Strukturen des Unternehmens verknüpfen (vgl. [SCS⁺02]).

Ein zweites Problem zeichnet sich durch einen Mangel an speziellen Methoden und Werkzeugen aus, welche die agentenorientierte Entwicklung von Steuerungssystemen unterstützen sollen. Die fehlende methodische Herangehensweise realisierter Projekte erschwert die Nachvollziehbarkeit der Entwicklungsschritte und beeinträchtigt die Wiederverwendbarkeit erreichter Teillösungen. Weitere Projekte müssen dann auf „grüner Wiese“ entstehen, was hohe Investitionskosten bedeutet und die Unternehmen vom Einsatz der Agententechnik abhält.

Unter diesem Gesichtspunkt erscheint die langfristige Reduktion des Entwicklungsaufwands als sinnvolle und anspruchsvolle Aufgabe. Die wissenschaftliche Herausforderung besteht dabei in der Besonderheit von Steuerungssystemen. Im Unterschied zu den meisten IT-Systemen ist die Steuerung mit einer physikalischen Anlage verknüpft. Dadurch ergeben sich spezielle Anforderungen an das Design und die Implementierung der agentenbasierten Softwaresysteme (vgl. [BJW04, S. 52]).

Darüber hinaus sorgt die Reduktion des Entwicklungsaufwands für eine schnellere Einführung von agentenbasierten Steuerungssystemen auf dem Markt. Aus ökonomischer Sicht sollen dabei die Entwicklungskosten bei neuen und insbesondere anschließenden Projekten gering gehalten werden, was dem Steuerungshersteller wettbewerbliche Vorteile ermöglicht.

1.2. Zielsetzung und Anforderungen

Ziel der vorliegenden Arbeit ist es, eine methodische Vorgehensweise zur Entwicklung von agentenbasierten Materialflusssteuerungen zu erarbeiten. Die Neuheit der Arbeit liegt dabei im Gegenstand der Betrachtung. Die zu erarbeitende Vorgehensweise soll eine Unterstützung bei der Entwicklung von Steuerungen für ortsfeste Stetigfördersysteme garantieren. Im Folgenden werden wesentliche Anforderungen an die Vorgehensweise formuliert (vgl. Abbildung 1.1).

Ziel der Vorgehensweise	Gegenstand der Betrachtung
• Entwicklung dezentraler Materialflusssteuerungen • Langfristige Reduktion des Entwicklungsaufwands	• Einsatz für die Steuerung von Stetigfördersystemen • Durchgehende Betrachtung aller Entwicklungsphasen • Agentenbasierte Materialflusssteuerung als Zielsystem
Inhaltliche Gütekriterien	Praktische Anwendbarkeit
• Entwicklung individueller Lösungen • Wiederverwendbarkeit bestehender Lösungen	• Benutzerorientierung • Erweiterbarkeit • Benutzerfreundlichkeit

Abbildung 1.1.: Anforderungen an die Vorgehensweise zur Entwicklung agentenbasierter Materialflusssteuerungen

Ziel der Vorgehensweise Die Vorgehensweise dient der Entwicklung dezentraler Materialflussteuerungen auf Basis von Softwareagenten. Mithilfe dieser Vorgehensweise soll die Entwicklung von Steuerungssystemen nachvollziehbar und wiederholbar gestaltet werden. Durch Verwendung definierter Entwicklungsschritte soll die Entwicklung neuer Systeme beschleunigt werden. Die Wiederverwendbarkeit einzelner Entwicklungsschritte soll bei einem mehrfachen Einsatz der Vorgehensweise zu einer langfristigen Reduktion des Entwicklungsaufwands beitragen.

Gegenstand der Betrachtung Das betrachtete Spektrum von zu steuernden technischen Anlagen begrenzt sich auf ortsfeste Stetigfördersysteme für Stückguttransporte in der Intralogistik. Als Basis für die Vorgehensweise dienen bestehende funktionale Anforderungen an die Anlage und deren Steuerung. Anhand dieser Vorgaben soll die Vorgehensweise den Entwickler in allen Phasen der Systementwicklung durchgehend unterstützen. Die resultierende Materialflussteuerung ist ein Multiagentensystem, in dem Softwareagenten als Gestaltungs- und Implementierungsmittel dienen. Die Vorgehensweise soll Erfahrungen aus der agentenorientierten Softwareentwicklung sowie existierende Agentenstandards berücksichtigen.

Inhaltliche Gütekriterien Inhaltliche Anforderungen an die Vorgehensweise lassen sich in zwei Punkten zusammenfassen. Einerseits soll die Entwicklung individueller Lösungen durch einen systematischen Aufbau der Vorgehensweise unter Verwendung eines konkreten methodischen Ansatzes beschleunigt werden. Andererseits soll die Wiederverwendbarkeit bereits vorhandener Lösungen und Teillösungen durch Verwendung von Klassifikationen, Modellen und Entwurfsmustern gewährleistet werden.

Praktische Anwendbarkeit Die praktische Anwendbarkeit einer Vorgehensweise hängt von Faktoren wie Verständlichkeit und Handhabbarkeit ab und trägt zur Akzeptanz der Vorgehensweise in der Praxis bei. Für die vorliegende Arbeit sind die Anforderungen an die praktische Anwendbarkeit in drei Kategorien aufgeteilt: Benutzerorientierung, Erweiterbarkeit und Benutzerfreundlichkeit. Durch die Benutzerorientierung wird eine bessere Akzeptanz unter Steuerungsentwicklern und IT-Spezialisten angestrebt. Zu diesem Zweck soll die Vorgehensweise auf Standards und Methoden der Automatisierung und Softwaretechnik aufbauen. Unter Erweiterbarkeit wird die Fähigkeit der Vorgehensweise verstanden, sich an verschiedene Anwendungsfälle und breit gestreute Anforderungen anzupassen. Unter Benutzerfreundlichkeit wird hier die Fähigkeit verstanden, den Entwicklungsprozess durch relevante Beispiele, Leitfragen und Softwarewerkzeuge zu unterstützen.

1.3. Vorgehensweise und Aufbau der Arbeit

Abbildung 1.2 zeigt eine allgemeine Vorgehensweise zur Entwicklung der Software einer Materialflusssteuerung. Bei dieser Vorgehensweise soll eine Vorstellung vom System in eine funktionsfähige Systemrealisierung überführt werden. Die Systemvorstellung wird in der Regel vom Kunden formuliert und weist eine Sichtweise auf, die sich vom Blickwinkel des Entwicklers unterscheiden kann. Hierdurch entsteht die erste semantische Lücke im Entwicklungsprozess. Um diese Lücke zu schließen, wird gerade im Bereich der Steuerung eine domänenspezifische Analyse und Modellierung des Problemfelds benötigt. Werden anschließend universale Methoden der Softwareentwicklung eingesetzt, so wird bei der Systemdefinition das Steuerungsproblem in Begriffen der Softwaretechnik formuliert und nach einem Systementwurf ein Implementierungsmodell erarbeitet. Die Herausforderung bei diesem Schritt besteht in der Anwendung einer allgemeinen Entwicklungsmethodik wie die objekt- oder agentenorientierte Softwareentwicklung auf die spezielle Problemdomäne der Materialflusssteuerung. Eine richtige Methodenauswahl, typische Modellierungsbeispiele oder andere relevante Empfehlungen können an dieser Stelle hilfreich sein, um dem Entwickler den Weg zum Zielsystem zu verkürzen. Die Überführung eines Implementierungsmodells in eine Software stellt eine weitere Herausforderung dar. Die zweite semantische Lücke entsteht dabei durch eine Diskrepanz zwischen den eingesetzten Modellierungsmitteln und dem resultierenden Programmcode. Abhilfe schaffen zum Teil spezielle Entwicklungsumgebungen und CASE-Werkzeuge¹.

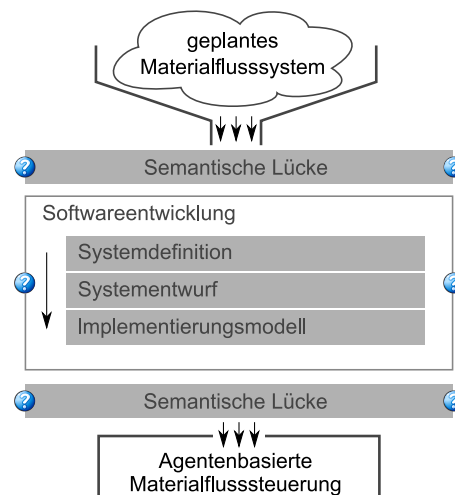


Abbildung 1.2.: Allgemeine Vorgehensweise zur Entwicklung einer Materialflusssteuerung

¹CASE ist die Abkürzung für Computer-aided Software Engineering.

Die hier beschriebenen Mängel einer allgemeinen Vorgehensweise sowie die Anforderungen aus ►Abschnitt 1.2 begründen folgenden Aufbau der Arbeit:

Die Ausgangssituation und die Motivation dieser Arbeit, werden in ►Kapitel 2 behandelt. Die Begründung der Aufgabenstellung wird aus dem Stand der Technik, den Defiziten konventioneller Materialflussteuerungen und den aktuellen Entwicklungstrends hergeleitet.

Die zu erarbeitende Vorgehensweise soll die Erstellung von agentenbasierten Materialflussteuerungen unterstützen. Aus diesem Grund werden die notwendigen Grundlagen der Agentenorientierung in ►Kapitel 3 zusammengefasst.

In ►Kapitel 4 wird ein problemspezifisches Anforderungsmodell für dezentrale Materialflussteuerungen aufgestellt. Bei der Beschreibung von Anforderungen wird, im Gegensatz zur klassischen Vorgehensweise, *von unten nach oben* vorgegangen. Dabei werden zuerst einzelne Systemelemente modelliert und erst danach komplexe Steuerungsaufgaben spezifiziert. Das angestrebte Anforderungsmodell soll hierbei die Ermittlung und Definition von Anforderungen unterstützen und damit die semantische Lücke zwischen den Kundenwünschen und dem Softwareentwurf (s. Abbildung 1.2) schließen.

In ►Kapitel 5 werden Maßnahmen erarbeitet, um den Aufwand beim Entwurf der Steuerungssoftware zu reduzieren. Dieser Teil der Arbeit verfolgt zwei Ziele: Zum einen soll die Entwicklung individueller Lösungen beschleunigt werden. Zum anderen soll die Wiederverwendbarkeit vorhandener Lösungen und Teillösungen verbessert werden. Dazu wird anstatt auf eine allgemeine objektorientierte Softwareentwicklungsmethode (s. Abbildung 1.2) auf eine geeignete agentenorientierte Entwicklungsmethode zurückgegriffen, welche für die Domäne der Materialflussteuerung angepasst wird.

Im Anschluss bietet ►Kapitel 6 Hilfestellungen, um die mithilfe der Agentenorientierung entwickelten Implementierungsmodelle zu lauffähigen Steuerungsprogrammen umsetzen zu können (s. Abbildung 1.2). Dabei wird der Entwickler bei der Auswahl der für die Implementierung geeigneten Werkzeuge unterstützt. Eine zusätzliche Unterstützung liefern Vorschläge zur Auswahl der Systemarchitektur sowie sinnvolle Varianten der Agentenverteilung.

Die Entwicklungsschritte in ►Kapitel 4, ►Kapitel 5 und ►Kapitel 6 bilden zusammen ein Vorgehensmodell, das für den praktischen Einsatz geeignet sein soll. Die Anwendbarkeit dieses Vorgehensmodells wird in ►Kapitel 7 anhand von drei Systembeispielen gezeigt, und die Eignung hinsichtlich der Anforderungen wird diskutiert.

►Kapitel 8 fasst die wesentlichen Ergebnisse der Arbeit zusammen und gibt einen Ausblick auf mögliche weiterführende Forschungsthemen.