

Erstellung von Steuerungssoftware für automatisierte Materialflusssysteme per Drag & Drop

Engineering the Control Software of Automated Material Handling Systems via Drag & Drop

Markus Spindler¹
Thomas Aicher²
Birgit Vogel-Heuser²
Johannes Fottner¹

¹ Lehrstuhl für Fördertechnik Materialfluss Logistik (fml)
Fakultät Maschinenwesen
Technische Universität München

² Lehrstuhl für Automatisierung und Informationssysteme
Fakultät Maschinenwesen
Technische Universität München

Automatisierte Materialflusssysteme ermöglichen hohe Durchsätze und kurze Durchlaufzeiten, benötigen jedoch eine individuell erstellte Steuerungssoftware, welche heutzutage überwiegend manuell erstellt wird. In diesem Beitrag wird für eine modulare Architektur zur Steuerungserstellung eine Basis-Bibliothek und ein Drag & Drop-Vorgehen zur Steuerungserstellung in einer grafischen Entwicklungsumgebung vorgestellt.

[Automatisierte Materialflusssysteme, Steuerungserstellung, Model-Driven Engineering]

Automated material handling systems allow high throughputs and short process times, but need individual control software, which is normally manually created. In this paper we present a basic library and drag & drop process, which allows the creation of the control software in a graphical editor without manual programming.

[Automated material handling systems, control software, model-driven software development]

1 EINFÜHRUNG

Automatisierte Materialflusssysteme (MFS) sind aufgrund ihrer kurzen Durchlaufzeiten und hohen Durchsätze in der Industrie weit verbreitet. Diese Systeme werden kundenspezifisch projektiert, weshalb eine individuelle Steuerungssoftware notwendig ist. Heutzutage erfordert das Erstellen dieser Steuerungssoftware umfangreiches manuelles Programmieren, da in der Regel eine modulare Steuerungssoftware mit vordefinierten Modulen fehlt. Im Rahmen des Forschungsprojekts aComA* – automatische

Codegenerierung für modular aufgebaute Anlagen in der Intralogistik – wird diese Lücke geschlossen, indem ein Konzept zur modularen Erstellung der Steuerungssoftware für hierarchisch gesteuerte MFS entwickelt wird.

2 STAND DER TECHNIK

Automatisierte MFS werden typischerweise durch eine zyklisch arbeitende Steuerung gesteuert, welche dezentral oder zentral ausgeführt sein kann. In verschiedenen Forschungsprojekten werden die Möglichkeiten von dezentralen Steuerungen betrachtet, da sie eine vergleichsweise hohe Flexibilität haben [Gün-2010; May-2009; Lib-2011]. Heutzutage werden im industriellen Umfeld jedoch überwiegend zentrale Steuerungen eingesetzt, da diese ein bewährtes Steuerungskonzept mit schneller Reaktionszeit des Gesamtsystems darstellen, weshalb sich diese Arbeit auf zentrale Steuerungen fokussiert.

Die Steuerungssoftware für automatisierte Systeme kann durch Model-Driven Engineering (MDE) effizient erstellt werden [Boh-2009]. Die Idee von MDE ist es, ein Modell der kundenindividuellen Anlage zu erstellen, um die Steuerungssoftware anhand dieses Modells automatisch zu generieren [Vog-2005]. Die Modellerstellung erfolgt hierbei typischerweise anhand vordefinierter Module, da dies einen zügigen Erstellungsprozess ermöglicht. Um solch ein Vorgehen verwenden zu können, muss für die betrachtete Domäne von automatisierten Systemen eine passende Systembeschreibung vorliegen. Für verschiedene Domänen von automatisierten Systemen existieren domänenspezifische Systembeschreibungen, beispielsweise Gebäudeautomation [Run-2011], verfahrenstechnische Anlagen [Ste-2012], Werkzeugmaschinen [Sch-2001] oder fertigungstechnische Anlagen [Güt-2012]. Für MFS

existiert aktuell keine Systembeschreibung, anhand derer die Steuerungssoftware erstellt werden kann, weshalb MDE in dieser Domäne aktuell nicht verwendet werden kann. In der Literatur finden sich jedoch Arbeiten, welche Aspekte von MFS beschreiben und dadurch als Grundlage zur Entwicklung einer Systembeschreibung verwendet werden können.

Zentral gesteuerte MFS haben üblicherweise eine Steuerungssoftware, die in Form einer sogenannten Automatisierungspyramide strukturiert ist. In dieser Pyramide werden verschiedene Steuerungsaufgaben auf verschiedenen Ebenen der Pyramide verteilt [Krä-2002]. Die Schnittstellen zwischen diesen Ebenen werden in der VDI-Richtlinie SAIL – System-Architektur für die Intra-logistik – abstrakt beschrieben [VDI-5100]. Diese beiden Elemente, die Aufteilung der Aufgaben auf verschiedene Ebenen und das Vordefinieren von Schnittstellen zwischen diesen Ebenen, werden in der hier verwendeten Systembeschreibung angewendet.

Bezüglich der Art der Modularisierung schlägt Wilke eine funktionsorientierte Modularisierung vor, in welcher jedes Modul eine Funktion kapselt [Wil-2006]. In Anlehnung daran wird in der hier vorliegenden Arbeit eine Bündelung von logistischen Funktionen in Modulen verwendet. Aicher et al. präsentiert ein Meta-Modell für MFS, welches ein MFS als Zusammensetzung einzelner Module beschreibt, die über vordefinierte Schnittstellen kommunizieren [Aic-2016; Reg-2016]. Hierbei werden komplexe logistische Funktionen aus einzelnen Operationen zusammengesetzt. Dies wird im hier vorliegenden Konzept ebenfalls verwendet.

Darüber hinaus werden in Forschungsarbeiten verschiedene Steuerungs-Architekturen für MFS entwickelt. Ten Hompel et al. stellt eine Zwei-Schichten-Architektur für MFS vor, in welcher die Ansteuerung der Hardware von den logistischen Entscheidungen getrennt wird [Hom-2006]. Diese Trennung der Fördermittel von den logistischen Entscheidungen wird in der vorliegenden Arbeit ebenfalls eingesetzt.

Des Weiteren stellen Lepuschitz et al. [Lep-2012] und Black und Vyatkin [Bla-2010] Architekturen für die Steuerungssoftware von MFS vor. Diese Arbeiten betrachten jedoch nur 1:1-Beziehungen zwischen Steuer-Modulen und den Fördermitteln und verwenden lediglich Informationen von direkt angrenzenden Modulen. Hierdurch ergeben sich bei der Anordnung der Module Einschränkungen, beispielsweise können keine Kreuzungen mit Gegenverkehr realisiert werden, weshalb nicht alle Anforderungen von MFS erfüllt sind.

Der Stand der Technik zeigt auf, dass MDE bereits zur Erstellung der Steuerungssoftware eingesetzt wird. Hierzu ist jedoch immer eine passende Systembeschreibung erforderlich, welche für MFS bisher nicht existiert.

3 BESTEHENDE VORARBEITEN UND INHALT DIESES BEITRAGS

Zwei-Schichten-Architektur

In vorhergehenden Arbeiten wurde ein modellbasierter Ansatz zur Erstellung der Steuerungssoftware entwickelt, der auf einer Zwei-Schichten-Architektur beruht [Spi-2016]. Diese Architektur trennt die Ansteuerung der Fördermittel von den Entscheidungen des Materialflusses, wodurch zwei Schichten entstehen: eine Logistik-Schicht, in welcher die logistischen Funktionen gesteuert werden (z. B. T-Verteilung) und eine Fördertechnik-Schicht, in welcher die Fördermittel gesteuert werden (siehe Abbildung 1). Die Fördertechnik-Schicht enthält für jedes Fördermittel genau ein Fördertechnik-Modul, welches die Sensorik und Aktorik einbindet. Die Steuerung der logistischen Funktionen in der Logistik-Schicht erfolgt durch Logistik-Module. Diese Module sind aus Transfer-Modulen aufgebaut, wobei jedes Transfer-Modul eine Operation ausführt (z. B. die kleinstmöglich kontrollierbare Quelle/Senke-Beziehung). Dieser Aufbau der Logistik-Module wird verwendet, um anhand weniger Transfer-Module die Vielzahl an logistischen Funktionen abzubilden.

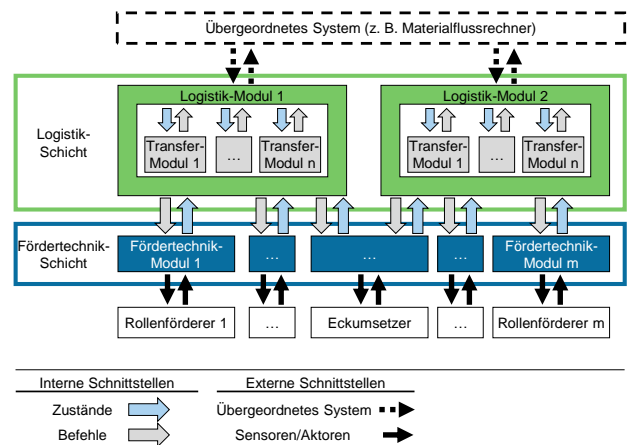


Abbildung 1. Zwei-Schichten-Architektur

Generischer Steuerungsalgorithmus

Der Steuerungsablauf dieser Architektur ist in einem generischen Steuerungsalgorithmus mit fünf Phasen definiert (siehe Abbildung 2) [Spi-2017]. Dieser Steuerungsalgorithmus beschreibt für jede Phase, welches Modul aufgerufen wird und welche Aufgabe das Modul bearbeitet: In Phase A generieren die Fördertechnik-Module anhand der ausgelesenen Sensoren den aktuellen Anlagenzustand. Auf Grundlage dieses Anlagenzustandes koordinieren die Logistik-Module in Phase B ihre enthaltenen logistischen Operationen (Transfer-Module). Die Transfer-Module steuern Phase C den Ablauf der logistischen Operationen und erzeugen hierfür die Befehle, welche an die Logistik-Module übergeben und dort in Phase D aggregiert werden. Die aggregierten Befehle werden

abschließend an die Fördertechnik-Module übergeben, welche in Phase E die Befehle anhand der Aktorik umsetzen. Der generische Steuerungsalgorithmus stellt eine Vorlage dar, anhand welcher spezifische Module ausgearbeitet werden können. Das Ausarbeiten der Module anhand dieser Vorlage stellt sicher, dass unabhängig voneinander ausgearbeitete Module miteinander kompatibel sind.

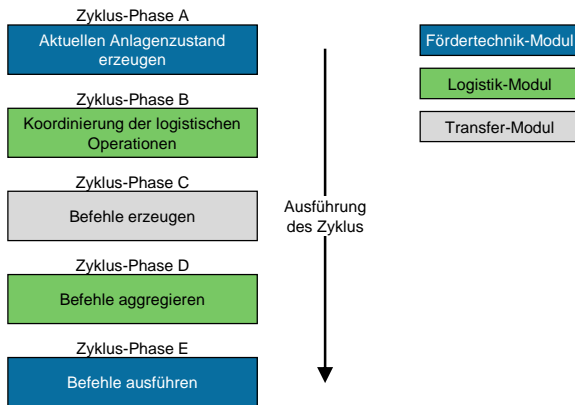


Abbildung 2. Phasen des generischen Steuerungsalgorithmus

In dem hier vorliegenden Beitrag werden für die typischerweise vorkommenden logistischen Funktionen und Fördermittel die Module ausgearbeitet. Damit wird zum einen aufgezeigt, dass anhand der entwickelten Architektur die Komplexität der Steuerung von industriell eingesetzten MFS abgedeckt ist und zum anderen wird eine Basis-Bibliothek ausgearbeitet, welche alle häufig eingesetzten Module enthält. Basierend auf den erarbeiteten Ergebnissen kann für den Großteil der in der Praxis relevanten Systemkonfigurationen anhand einer grafischen Entwicklungsumgebung per Drag & Drop die Steuerungssoftware erstellt werden.

4 ANALYSE TYPISCHER ELEMENTE IN MFS

Hinsichtlich der logistischen Funktion Fördern, z. B. staufähiges Fördern und nicht staufähiges Fördern, und der verwendeten Vorfahrts- und Verteilstrategien für die logistische Funktion Zusammenführen/Verteilen gibt es umfangreiche Literatur [Jün-2000; Gün-2017; Gud-2012]. Jedoch finden sich in der Literatur keine Informationen bezüglich der typischerweise verwendeten Kreuzungen und der Häufigkeit der verwendeten Fördermittel und Handhabungsgeräte, weshalb diese analysiert wurden.

Für die Analyse der typischerweise vorkommenden Kreuzungsstrukturen wurden die enthaltenen Kreuzungstypen in 23 Layouts (12 Behälterfördertechnik und 11 Palettenfördertechnik) von sieben unterschiedlichen Anbietern analysiert. Hierfür wurden in jedem Layout die enthaltenen Kreuzungen (insgesamt 716 Kreuzungen) identifiziert und für die unterschiedlichen Kreuzungstypen die Anzahl gezählt. Das Ergebnis dieser Analyse ist, dass

die T-Kreuzung mit 67 % der am häufigsten vorkommende Kreuzungstyp ist. Insgesamt können mit sieben Kreuzungstypen 95 % der Kreuzungen abgedeckt werden und lediglich 5 % der Kreuzungen sind individuelle Kreuzungen, welche in der Kategorie Sonstige gebündelt sind (siehe Abbildung 3).

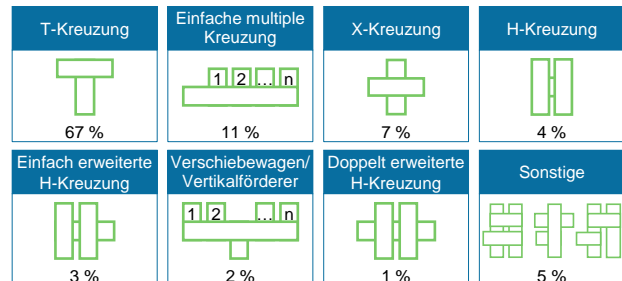


Abbildung 3. Analytierte prozentuale Häufigkeit der Kreuzungstypen in MFS

Für die Analyse der typischerweise verwendeten Fördermittel und Handhabungsgeräte wurden in 17 Layouts (12 Paletten und 5 Behälter) von fünf unterschiedlichen Anbietern mit insgesamt 3.282 Fördermitteln (1.818 Palettenfördertechnik und 1.464 Behälterfördertechnik) und 59 Handhabungsgeräten die Häufigkeit der Typen der Fördermittel und Handhabungsgeräte untersucht. Hieraus ergab sich, dass mit fünf Arten von Fördermitteln bei der Behälterfördertechnik 95,4 % der Fördermittel und bei der Palettenfördertechnik 98,5 % der Fördermittel abgedeckt sind (siehe Abbildung 4). Darüber hinaus ergaben sich die häufigsten Handhabungsgeräte Konturenkontrolle (42 %), Waage (17 %), automatisierte Magazine für Paletten sowie Behälter (16 %) und Kufen- und Fußfreiraumkontrolle (12 %). Die restlichen 13 % der Handhabungsgeräte verteilen sich auf sonstige Geräte, z. B. Belegbegeber, Etikettierer oder Palettenwickler.

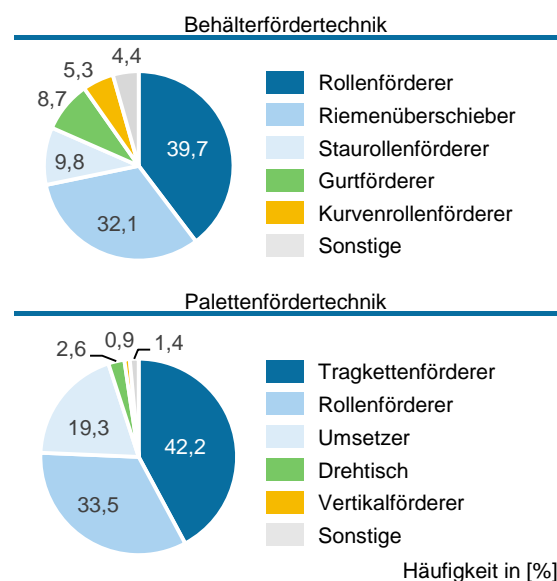


Abbildung 4. Analytierte prozentuale Häufigkeit der verwendeten Fördermittel in MFS

Diese häufig vorkommenden Module bilden die Basis-Bibliothek, welche im Rahmen dieser Arbeit ausgearbeitet wird. Um in MFS zusätzliche Funktionen zu erhalten, werden die Fördermittel mit Features ausgestattet. Solche Features wurden mit Experten aus dem Bereich der MFS im Rahmen von Interviews diskutiert und es wurden fünf typisch eingesetzte Features identifiziert, welche ebenfalls in die Basis-Bibliothek aufgenommen werden:

- Unterschiedliche Lichtschranken-Konfigurationen zum Detektieren von Transporteinheiten,
- Verwendung von virtuellen Sensoren,
- Einbinden von Schaltern,
- Optimierungs-Lichtschranken und
- Einbindung eigenständig gesteuerter Fördermittel.

Unterschiedliche Lichtschranken-Konfigurationen zum Detektieren von Transporteinheiten

Für das Detektieren von Transporteinheiten wird üblicherweise eine einzige Lichtschranke verwendet. In bestimmten Fällen werden jedoch mehrere Lichtschranken mit einer UND-Verknüpfung bzw. ODER-Verknüpfung oder Lichtschranken zur Spaltkontrolle eingesetzt, beispielsweise um bei Kreuzungselementen Transporteinheiten aus verschiedenen Richtungen mittig auf dem Kreuzungselement zu stoppen.

Verwendung von virtuellen Sensoren

An manchen Stellen im MFS wird der Zustand der Anlage anhand von virtuellen Sensoren, welche auf Basis von Timern arbeiten, bestimmt. Beispielsweise kann mit virtuellen Sensoren bei einem Riemenüberschieber die obere Hub-Position der Riemen überwacht oder eine Transporteinheit, welche zum Ausrichten gegen einen Anschlag gefahren wird, detektiert werden.

Einbindung von Schaltern

Schalter werden in MFS eingebunden, um die Steuerung manuell zu beeinflussen. Hierbei werden typischerweise An-Aus-Schalter mit zwei möglichen Schalterstellungen verwendet. Ein An-Schalter wird beispielsweise verwendet, wenn durch einen manuellen Prozess (z. B. Stapler) eine Palette in das MFS eingeschleust wird, um dem System mitzuteilen, dass der Abtransport der Palette beginnen darf. Ein Aus-Schalter kann eingesetzt werden, wenn z. B. durch einen Schalter ein Übertrittsbereich vorübergehend für den Transport gesperrt und anschließend wieder freigegeben werden soll.

Optimierungs-Lichtschranken

Optimierungs-Lichtschranken werden an Stellen im MFS eingebaut, die kritisch bezüglich des Durchsatzes

sind. Das Ziel einer Optimierungs-Lichtschranke ist, ein Fördermittel früher für eine neue Aufgabe freizugeben und dadurch den Durchsatz des entsprechenden System-Teils zu erhöhen.

Einbindung eigenständig gesteuerter Fördermittel

Für das staufähige Fördern existieren Fördermittel, welche die logistische Funktion eigenständig ausführen. Hierfür besitzen diese einen Mikrocontroller, der die Sensorik ausliest und die Aktorik entsprechend ansteuert. Diese Fördermittel werden eingesetzt, da sie den Aufwand der Inbetriebnahme für die Anlage reduzieren und die SPS hinsichtlich der Rechenleistung entlasten.

5 AUSARBEITUNG DER MODULE

Die Module der Basis-Bibliothek werden im Rahmen dieser Arbeit ausgearbeitet. Die detaillierte Ausgestaltung dieser Module erfolgt hierbei anhand des zuvor beschriebenen generischen Steuerungsalgorithmus.

Um die Module korrekt auszuarbeiten, werden zuerst in Abstimmung mit Experten aus der Industrie für die logistischen Funktionen Anforderungen definiert. Bei einer ruhenden Handhabungsaufgabe (z. B. Palettenwickler) umfassen diese Anforderungen beispielsweise folgende sechs Punkte:

- Handhabungsgerät muss ausgeschaltet sein, wenn es nicht verwendet wird,
- Transporteinheit darf nur zum Handhabungsgerät gefördert werden, wenn das Handhabungsgerät bereit ist,
- Bestimmung der Programmnummer in Abhängigkeit der ID der Transporteinheit
- Nicht identifizierbare Transporteinheiten (No-Read) werden ohne Handhabung auf einen Default-Weg geschickt,
- Start der Handhabung, sobald die Transporteinheit angehalten ist und
- Abfordern der Transporteinheit, nachdem die Handhabung abgeschlossen ist

Aufbauend auf diese festgelegten Anforderungen werden die spezifischen Module ausgearbeitet. Hierzu werden für die drei Modularten (Fördertechnik-Module, Logistik-Module, Transfer-Module) zuerst die Prozesse definiert und anschließend formalisiert beschrieben, um danach die Module implementieren zu können. Dieses Vorgehen wird nachfolgend für die drei Modularten erläutert.

Fördertechnik-Module

Beim Erstellen der Fördertechnik-Module sind zwei wesentliche Aspekte auszuarbeiten: das Einbinden der spezifischen Sensorik/Aktorik und das Abbilden des Funktionsprinzips des Fördermittels. Beim Einbinden der spezifischen Sensorik und Aktorik wird festgelegt, welche Sensorik und Aktorik das Fördermittel besitzt und wie diese eingebunden wird. Bei der Sensorik umfasst dies die Datentypen und Wertebereiche der Messwerte (inkl. deren Bedeutung) und bei der Aktorik die Art und Anzahl der Aktoren und welche Stellsignale diese jeweils benötigen. Hinsichtlich des Abbildens des Funktionsprinzips des Fördermittels wird beschrieben, welche Aktionen das Fördermittel in welcher Position durchführen kann, z. B. die Abhängigkeit des Drehwinkels und der möglichen Förderrichtungen bei einem Drehtisch oder dass die Handhabung eines Palettenwicklers nur aus der Grundstellung heraus gestartet werden kann.

Transfer-Module

Bei den Transfer-Modulen wird zuerst der Prozessablauf der Operation definiert. Dies umfasst, welche Aktionen und Bedingungen in welcher Reihenfolge für das Ausführen der Prozesse erfüllt sein müssen. Bei der Handhabung eines Palettenwicklers umfasst dies beispielsweise, dass zunächst eine Programmnummer eingegeben muss, bevor die Handhabung gestartet werden kann oder dass das Handhabungsgerät nach dem Beenden der Handhabung wieder in die Grundstellung fahren muss. Dieser Prozessablauf inklusive der Übergangsbedingungen wird in Form eines Zustandsdiagramms beschrieben.

Logistik-Module

Der Kern bei der Beschreibung der Logistik-Module ist, wie die enthaltenen Transfer-Module aufeinander abgestimmt werden, also welche Abhängigkeiten zwischen den enthaltenen Transfer-Modulen bestehen. Bei Zusammenführungen umfasst dies das Vorfahrtsverhalten, bei einer Verteilung die Entscheidung, in welche Richtung eine Transporteinheit gefördert wird, und bei einem Palettenwickler wird hierdurch beispielsweise festgelegt, dass das Einfördern erst stattfinden darf, wenn drei Bedingungen erfüllt sind: Die Handhabungsoperation muss komplett abgeschlossen sein (Handhabungsgerät ist in Grundposition), eine Transporteinheit zum Einfördern ist vorhanden und der Stellplatz im Handhabungsgerät ist frei. Diese Zusammenhänge werden in Form von Zustandsdiagrammen beschrieben.

Integration von Features

Um die Features in die Module zu integrieren, werden die jeweiligen Module um die gewünschten zusätzlichen Funktionen erweitert. Hierzu wird beispielsweise den Fördertechnik-Modulen die Option hinterlegt, entweder eine oder mehrere Lichtschranken zur Detektion von

Transporteinheiten einzulesen oder die Transfer-Module werden entsprechend des erweiterten Prozessablaufs ergänzt, um beispielsweise Optimierungs-Lichtschranken umzusetzen.

Anhand dieses Vorgehens werden für die spezifischen Module die formalisierten Dokumente erstellt, welche die Grundlage für die anschließende Implementierung der Module sind. Durch diese Trennung der Modulbeschreibung von der Implementierung wird die Gesamtkomplexität der Modulentwicklung in Teilschritte mit jeweils geringerer Komplexität zerlegt, wodurch das Entwickeln zügiger und einfacher durchgeführt werden kann. Die entwickelten Module werden im nächsten Schritt getestet, um ihre Funktion zu verifizieren.

6 VERIFIZIERUNG DER MODULE

Die Verifizierung der Module erfolgt in einem zweistufigen Vorgehen: Zuerst werden die einzelnen Module Modultests unterzogen und anschließend wird ein Integrationstest durchgeführt, bei dem das Zusammenwirken der Module verifiziert wird.

Anhand der Modultests wird getestet, ob die implementierten Module die anfangs aufgestellten Anforderungen erfüllen. Hierzu werden Testlayouts erstellt, in welchen die Modulfunktion isoliert betrachtet wird. Für den Modultest des Palettenwicklers umfasst dies beispielsweise eine Quelle, Rollenförderer für den Transport zum Palettenwickler, den Palettenwickler inkl. eines Identifizierungsgeräts und Rollenförderer für den Abtransport inkl. einer Senke (siehe Abbildung 5). Diese Testlayouts werden in einer Emulationsumgebung aufgebaut und anhand der entwickelten Module gesteuert. Durch die Visualisierung der Emulationsumgebung und das Protokollieren der Sensor- und Aktorsignale können dann die Anforderungen verifiziert werden.

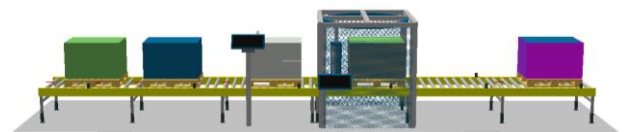


Abbildung 5. Layout für den Modultest des Palettenwicklers

Im Anschluss an die erfolgreichen Modultests erfolgt der Integrationstest, anhand welchem gezeigt wird, dass die Module flexibel zu einem individuellen Layout zusammengefügt werden können. Hierzu wird ein industrielles Layout der Palettenfördertechnik mit zwei Aufgabestationen, fünf Regalbediengeräten, vier Arbeitsplätzen und drei Bereitstellbahnen verwendet (siehe Abbildung 6). Insgesamt enthält dieses Layout 21 Kreuzungen, 22 Barcodescanner, fünf Arten von Handhabungsgeräten (Konturenkontrolle, automatisches Palettenmagazin, Palettenwickler, Etikettierer, Schnelllauftor) und verwendet als Fördermittel Kettenförderer (93 Stück), Rollenförderer (68 Stück) und Eckumsetzer (38 Stück).

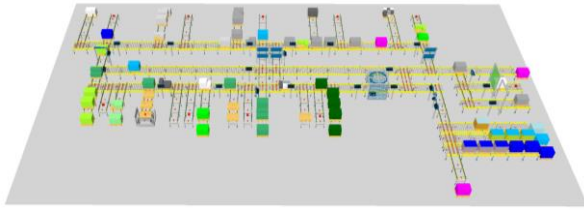


Abbildung 6. Industrielles Layout für den Integrationstest

Der überwiegende Teil der Steuerung dieses Layouts kann aus den Modulen der Basis-Bibliothek erstellt werden. Lediglich für drei Kreuzungen müssen projektspezifische Logistik-Module entwickelt werden. Diese Kreuzungs-Logistik-Module können aus den bereits existierenden Transfer-Modulen erstellt werden, wodurch der Erstellungsaufwand gering ist. Nach dem Erstellen der projektspezifischen Kreuzungs-Logistik-Module kann die Steuerungssoftware anhand der vorgefertigten Module erstellt werden. Um die so erstellte Steuerungssoftware zu testen, wird das Layout in einer Emulationsumgebung aufgebaut. Beim Ansteuern der Emulation zeigt sich, dass die Transporte und Handhabungen wunschgemäß durchgeführt werden: Die an den Senken erzeugten Paletten (Aufgabebereich, Auslagerstich der Regalbediengeräte, Kommissionierarbeitsplätze) werden in Abhängigkeit ihrer ID an den Kreuzungen zu den gewünschten Senken (Einlagerstich der Regalbediengeräte, Kommissionierarbeitsplatz, Nicht-In-Ordnung-Platz, Bereitstellbahn) transportiert. Beim Passieren von Handhabungsgeräten werden diese Geräte aktiviert und die für die jeweilige ID erforderliche Handhabung wird durchgeführt. Somit ist gezeigt, dass die Module flexibel miteinander kombiniert werden können und der Integrationstest ist bestanden. Dieses modulare Konzept der Steuerungssoftware wird in eine grafische Entwicklungsumgebung überführt, um die Steuerungssoftware per Drag & Drop zu erstellen.

7 GRAFISCHE ENTWICKLUNGSUMGEBUNG

Die grafische Entwicklungsumgebung ist in drei Abschnitte unterteilt (siehe Abbildung 7): Projektbaum (links), Arbeitsbereich (mittig) und Modulbibliotheken (rechts). Im ersten Schritt der Steuerungserstellung wird anhand der Fördertechnik-Module das Layout der Anlage nachgebaut. Hierzu werden die jeweiligen Fördertechnik-Module aus der Bibliothek ausgewählt und per Drag & Drop im Arbeitsbereich entsprechend des Layouts angeordnet. Nach dem Anordnen der Fördertechnik-Module erfolgt das Platzieren der Logistik-Module. Hierzu werden die Logistik-Module aus der entsprechenden Bibliothek ausgewählt und per Drag & Drop über den Fördertechnik-Modulen platziert. Anschließend werden die Logistik-Module anhand des Projektbaums mit den zugehörigen Fördertechnik-Modulen referenziert. Um in den Logistik-Modulen die gewünschte Strategie zu erhalten, wird das jeweilige Logistik-Modul ausgewählt und in den Einstellungen (unten rechts) wird die gewünschte Strategie per Parameter eingestellt.

Durch diese Entwicklungsumgebung in Kombination mit der ausgearbeiteten Basis-Bibliothek ist ein effizientes und intuitives Vorgehen für die Steuerungserstellung per Drag & Drop gegeben. Die Umsetzung der Zwei-Schichten-Architektur auf industrielle Förder- und Steuerungstechnik in IEC 61131-3 und die Berücksichtigung des grafischen Editors wurde bereits von Aicher et al. vorgenommen [Aic-2017].

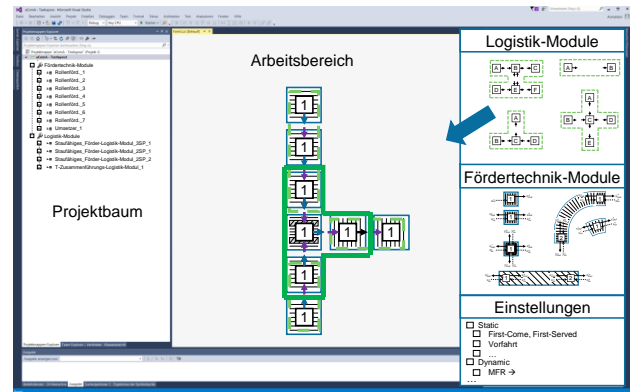


Abbildung 7. Schematische Darstellung der Entwicklungsumgebung

8 ZUSAMMENFASSUNG UND AUSBLICK

Aufgrund ihrer hohen Durchsätze und kurzen Durchlaufzeiten sind automatisierte MFS in der Industrie weit verbreitet. Diese Systeme werden kundenspezifisch projektiert und benötigen daher eine individuelle Steuerungssoftware. Heutzutage erfordert das Erstellen dieser Software umfangreiche manuelle Programmieraktivitäten, welche arbeitsintensiv und fehleranfällig sind. In diesem Beitrag wird anhand einer quantitativen Analyse gezeigt, dass bei MFS mit wenigen Modulen ein Großteil der Strukturen abgedeckt werden kann. Auf Grundlage dieser Analyse wird eine Basis-Bibliothek mit den häufigsten Modulen definiert. Diese Module der Basis-Bibliothek werden ausgearbeitet und verifiziert. Anschließend wird erläutert, wie anhand dieser vorgefertigten Module und einer grafischen Entwicklungsumgebung die Steuerungssoftware per Drag & Drop erstellt werden kann.

Aufbauend auf diese Ergebnisse können im nächsten Schritt die entwickelten Module und die grafische Entwicklungsumgebung in eine industrielle Implementierung überführt werden. Um die Entwicklungsumgebung weiter zu verbessern, können ihr wissensbasierte Elemente hinterlegt werden, welche die Erstellung der Steuerungssoftware unterstützen bzw. (teil-)automatisieren. Solche Elemente sind beispielsweise das automatisierte Anordnen der Fördertechnik-Module anhand einer eingelesenen AutoCAD Layout-Datei oder das Prüfen der Steuerungssoftware auf Inkonsistenzen, z. B. nicht erreichbare Senken oder das fehlerhafte Abstimmen einer Gewichtskontrolle auf das zulässige Gewicht eines Lagersystems.

LITERATUR

- [Aic-2016] Aicher, T.; Regulin, D.; Schütz, D.; Lieberoth-Leden, C.; Spindler, M.; Günthner, W. A.; Vogel-Heuser, B.: Increasing flexibility of modular automated material flow systems – A meta model architecture. In: Proceedings of the 8th IFAC Conference on Manufacturing Modelling Management and Control (MIM) (2016), S. 1543–1548.
- [Aic-2017] Aicher, T.; Dötsch, D.; Spindler, M.; Fottner, J.; Vogel-Heuser, B.: A feasibility study regarding the implementation and application of a two-layer architecture for automated material flow systems (accepted). In: Proceedings of the 43rd IEEE Annual Conference of the IEEE Industrial Electronics Society (IES) (2017)
- [Bla-2010] Black, G.; Vyatkin, V.: Intelligent Component-Based Automation of Baggage Handling Systems With IEC 61499. In: IEEE Transactions on Automation Science and Engineering, Jg. 7 (2010) Nr. 2, S. 337–351.
- [Boh-2009] Bohner, S. A.; Mohan, S.: Model-Based Engineering of Software: Three Productivity Perspectives. In: Proceedings of the 33rd Annual IEEE Software Engineering Workshop (SEW) (2009), S. 35–44.
- [Gud-2012] Gudehus, T.: Logistik 1 – Grundlagen, Verfahren und Strategien. Springer, Berlin, 2012.
- [Gün-2010] Günthner, W. A.; Hompel, M. ten (Hrsg.): Internet der Dinge in der Intralogistik. Springer, Berlin, Heidelberg, 2010.
- [Gün-2017] Günthner, W. A.: Förder- und Materialflusstechnik. Vorlesungsskriptum. Lehrstuhl für Fördertechnik Materialfluss Logistik, Technische Universität München, Garching, 2017.
- [Güt-2012] Güttel, K.: Konzept zur Generierung von Steuerungscode für Fertigungsanlagen unter Verwendung wissens-
- basierter Methoden. Dissertation. Professur für Automatisierungstechnik, Helmut-Schmidt-Universität Hamburg, Hamburg, 2012.
- [Hom-2006] Hompel, M. ten; Libert, S.; Sondhof, U.: Dezentrale Steuerung für Materialflusssysteme am Beispiel von Stückgutförder- und sortieranlagen. In: Logistics Journal (2006)
- [Jün-2000] Jünemann, R.; Schmidt, T.: Materialflußsysteme – Systemtechnische Grundlagen ; 36 Tabellen. Springer, Berlin, Heidelberg [u.a.], 2000.
- [Krä-2002] Krämer, K.: Automatisierung in Materialfluss und Logistik – Ebenen, Informationslogistik, Identifikationssysteme, intelligente Geräte. Deutscher Universitäts-Verlag, Wiesbaden, 2002.
- [Lep-2012] Lepuschitz, W.; Jirkovsky, V.; Kadera, P.; Vrba, P.: A Multi-Layer Approach for Failure Detection in a Manufacturing System Based on Automation Agents. In: Proceedings of the 9th International Conference on Information Technology: New Generations (ITNG) (2012), S. 1–6.
- [Lib-2011] Libert, S.: Beitrag zur agentenbasierten Gestaltung von Materialflusssteuerungen. Dissertation. Lehrstuhl für Förder- und Lagerwesen, Technische Universität Dortmund, Dortmund, 2011.
- [May-2009] Mayer, S. H.: Development of a completely decentralized control system for modular continuous conveyor systems. Dissertation. Institut für Fördertechnik und Logistiksysteme, Karlsruher Institut für Technologie, Karlsruhe, 2009.
- [Reg-2016] Regulin, D.; Aicher, T.; Vogel-Heuser, B.: Improving Transferability Between Different Engineering Stages in the Development of Automated Material Flow Modules. In: IEEE Transactions on Automation Science and Engineering, Jg. 13 (2016) Nr. 4, S. 1422–1432.
- [Run-2011] Runde, S.; Fay, A.: Software Support for Building Automation Re-

quirements Engineering – An Application of Semantic Web Technologies in Automation. In: IEEE Transactions on Industrial Informatics, Jg. 7 (2011) Nr. 4, S. 723–730.

Dissertation. Lehrstuhl für Förder-
technik Materialfluss Logistik,
Technische Universität München,
Garching, 2006.

- [Sch-2001] Schwenke, M. J.; Staron, R. J.: Use of a type library to speed up PLC program design and commissioning. In: Proceedings of the 2001 ISA Conference and Exhibition (2001), S. 381–393.
- [Spi-2016] Spindler, M.; Aicher, T.; Schütz, D.; Vogel-Heuser, B.; Günthner, W. A.: Modularized Control Algorithm for Automated Material Handling Systems. In: Proceedings of the 19th IEEE International Conference on Intelligent Transportation Systems (ITSC) (2016), S. 2644–2650.
- [Spi-2017] Spindler, M.; Aicher, T.; Habenicht, S.; Vogel-Heuser, B.; Fottner, J.: Generischer Steuerungs-Algorithmus zur Erstellung von Modulen für die Steuerungssoftware von ortsfesten Materialflusssystemen (submitted). In: Forschung im Ingenieurwesen (2017)
- [Ste-2012] Steinegger, M.; Zoitl, A.: Automated code generation for programmable logic controllers based on knowledge acquisition from engineering artifacts: Concept and case study. In: Proceedings of the 17th IEEE Conference on Emerging Technologies and Factory Automation (ETFA) (2012)
- [VDI-5100] Verein Deutscher Ingenieure: Systemarchitektur für die Intralogistik (SAIL) - Blatt 1. VDI/VDMA-Richtlinie Nr. 5100 Blatt 1, 2011.
- [Vog-2005] Vogel-Heuser, B.; Witsch, D.; Katze, U.: Automatic Code Generation from a UML model to IEC 61131-3 and system configuration tools. In: Proceedings of the 5th IEEE International Conference on Control & Automation (ICCA) (2005), S. 1034–1039.
- [Wil-2006] Wilke, M.: Wandelbare automatisierte Materialflusssysteme für dynamische Produktionsstrukturen.

Dipl.-Ing. Markus Spindler, Wissenschaftlicher Mitarbeiter am Lehrstuhl für Fördertechnik Materialfluss Logistik (fml) der Technischen Universität München.

Thomas Aicher, M. Sc., Wissenschaftlicher Mitarbeiter am Lehrstuhl für Automatisierung und Informationssysteme (AIS) der Technischen Universität München.

Prof. Dr.-Ing. Birgit Vogel-Heuser, Leiterin des Lehrstuhls für Automatisierung und Informationssysteme (AIS) der Technischen Universität München.

Prof. Dr.-Ing. Johannes Fottner, Leiter des Lehrstuhls für Fördertechnik Materialfluss Logistik (fml) der Technischen Universität München.

Adresse:

Technische Universität München, Lehrstuhl fml,
Boltzmannstr. 15, 85748 Garching, Deutschland,
Phone: +49 89 289 15417

E-Mail: spindler@fml.mw.tum.de