

Entwicklung eines Bausteinkastens zur Simulation mobiler Lagerroboter

Development of an Object Library for the Simulation of Robotic Mobile Fulfilment Systems

Kai Gutenschwager, Ostfalia Hochschule für angewandte Wissenschaften,
Wolfenbüttel (Germany), k.gutenschwager@ostfalia.de

Sven Völker, Hochschule Ulm, Ulm (Germany), voelker@hs-ulm.de

Abstract: Operating Robotic Mobile Fulfilment Systems (RMFS) requires to solve several interdependent and NP-hard planning problems. Recently, algorithms for controlling RMFS have been moved into the focus of research. Simulation is required to analyse the performance of RMFS under different control strategies. Hence, an adequate simulation environment is needed. We propose a concept of a simulation framework and present a realisation of the framework in Plant Simulation. The novelty of our approach is to enable an efficient modelling and management of alternative planning processes within a general warehouse management and a general robot control system, both on a global and a local level.

1 Einleitung

Der seit Jahren anhaltende Erfolg des Versandhandels zieht eine wachsende Zahl von Logistikzentren nach sich. Die Prozesse in diesen Logistikzentren werden aus verschiedenen Gründen, z. B. der Formenvielfalt der Artikel, nicht vollständig automatisiert. Robotic Mobile Fulfilment Systems versprechen eine Verbindung der Vorteile von automatischer und manueller Kommissionierung (Wurman et al. 2007, Enright und Wurman 2011). Sie verzichten weitgehend auf fest installierte Lager- und Fördertechnik. Stattdessen werden die Artikel in transportablen Regalen gelagert, die von mobilen Robotern zu den Kommissionierplätzen transportiert werden. Dort werden die Artikel manuell entnommen und verpackt. Charakteristisch für RMFS sind Layouts auf Basis rechtwinkliger Raster (Nobbe 2015).

Bei Auslegung und Betrieb von RMFS muss eine Reihe interdependenter Probleme gelöst werden (Emde und Boysen 2017, Merschformann et al. 2018b), wie beispielsweise die Festlegung des Lagerlayouts, die Sequenzierung von Aufträgen, die Allokation von Ressourcen und die Wegeplanung. Durch praktische Anforderungen, z. B. die Echtzeitforderung, wird die Lösung dieser Probleme zusätzlich erschwert.

Obwohl für jedes Teilproblem Lösungsverfahren existieren, ist das Gesamtproblem keineswegs abschließend gelöst und bedarf weiterer Forschung (Emde und Boysen 2017). Dennoch wurden bereits einige RMFS realisiert: Wulfrat (2012) gibt eine Übersicht über zwanzig RMFS-Implementierungen, die teilweise mehrere tausend Regale und einige hundert Roboter umfassen. Hanson et al. (2018) beschreiben ein RMFS mit 1.550 Regalen, 65 Robotern und 30.000 Lagereinheiten. Die in diesen RMFS eingesetzten Steuerungsverfahren sind dabei nicht detailliert erklärt.

Boysen et al. (2017) zeigen für ein Teilproblem, dass ein intelligentes Steuerungsverfahren die erforderliche Flottengröße gegenüber einer einfachen, regelbasierten Disposition halbieren kann. Das legt die Vermutung nahe, dass hinsichtlich der Lösungsgüte der Verfahren auch in der Praxis noch Verbesserungspotential besteht.

Die Leistungsfähigkeit von Verfahren zur Steuerung von RMFS kann analytisch nur eingeschränkt beurteilt werden. Für bedienungstheoretische Ansätze sei auf Yuan und Gong (2016 und 2017) sowie Lamballais et al. (2017) verwiesen. Um einzelne Steuerungsansätze in ihrem Zusammenspiel zu bewerten, ist Simulation ein geeigneter Ansatz. Aktuell am Markt verfügbaren Simulationswerkzeuge bieten jedoch keine Bausteine, die zur effizienten und realitätsnahen Modellierung von RMFS geeignet sind. Dieser Beitrag stellt ein Konzept für einen derartigen Bausteinkasten sowie eine Umsetzung des Konzepts für das Simulationswerkzeug Plant Simulation vor. Mit Hilfe des Bausteinkastens können Algorithmen zur Steuerung von RMFS mit relativ geringem Aufwand implementiert und getestet werden.

Abschnitt 2 beschreibt Planungsaufgaben im Kontext von RMFS, stellt mögliche Modellierungsansätze vor und gibt einen Überblick über verfügbare Software. In Abschnitt 3 wird ein Fachkonzept für den RMFS-Bausteinkasten entwickelt. Abschnitt 4 geht auf ausgewählte Aspekte der Realisierung des Bausteinkastens in Plant Simulation ein. Abschnitt 5 enthält Zusammenfassung und Ausblick.

2 Steuerung und Simulation von RMFS

Im laufenden Betrieb von RMFS sind verschiedene Planungsaufgaben zu erfüllen, wobei die Lösungsansätze in der betrieblichen Praxis in der Regel Prozesscharakter besitzen und unterschiedlichen Steuerungsebenen zugeordnet werden.

2.1 Planungsaufgaben und Lösungsansätze

Auf Ebene der Lagerverwaltung sind folgende Aufgaben angesiedelt:

1. Sequenzierung der eingehenden Kommissionieraufträge
2. Zuordnung der Kommissionieraufträge zu jeweils einem Kommissionierplatz
3. Zuordnung der Kommissionierauftragspositionen zu Regalen (Bilden der Transportaufträge)
4. Sequenzierung der Transportaufträge
5. Festlegung des Stellplatzes, auf dem ein Regal nach Erfüllung des Transportauftrags abzustellen ist

Die weiteren Planungsaufgaben sind der Roboterverwaltung zuzuordnen:

6. Zuordnung der Transportaufträge zu jeweils einem Roboter
7. Wegeplanung für die Roboter (Bilden der Fahraufträge)
8. Nachziehen und Umsetzen der vor Kommissionierplätzen wartenden Roboter

9. Verwaltung von Robotern ohne aktive Fahraufträge

Alle Planungsaufgaben sind unter Echtzeitbedingungen für den Online-Fall und unter Berücksichtigung stochastischer Größen zu lösen.

Zur Lösung der Planungsaufgaben 1 bis 6 werden in der Literatur Regelsysteme, Optimierungsverfahren und Auktionsverfahren vorgeschlagen (siehe z. B. Boysen et al. 2017 und Wurman et al. 2007). Die Schwierigkeit von Aufgabe 7 resultiert aus der Notwendigkeit, mögliche Verklemmungen zu vermeiden oder aufzulösen. Zudem ist zu berücksichtigen, dass beladene Roboter Regale nicht unterfahren können. Geeignete Verfahren zur Wegeplanung werden von Lienert und Fottner (2017) sowie Merschformann et al. (2018a) beschrieben. Planungsaufgabe 8 ergibt sich aus der Tatsache, dass Regale nicht unbedingt in der Reihenfolge am Kommissionierplatz eintreffen, in der sie benötigt werden. Dadurch wird ein gezieltes Nachziehen der wartenden Roboter an den Kommissionierplatz notwendig. Zu den Details realer Implementierungen von RMFS-Steuerungen finden sich bisher keine weiterführenden Veröffentlichungen.

Grundsätzlich können die Planungsaufgaben sequentiell oder (teilweise) simultan bearbeitet werden. Erfolgt die Problemlösung sequentiell, so werden *Planungsprozesse*, d. h. Folgen einzelner *Planungsschritte*, durchlaufen. Die Planungsprozesse werden im laufenden Betrieb durch unterschiedliche *Ereignisse* und somit zu verschiedenen Zeitpunkten ausgelöst. So kann beispielsweise die Sequenzierung eingehender Kommissionieraufträge immer bei Bekanntgabe eines neuen Auftrags erfolgen, die Zuordnung zu Kommissionierplätzen aber erst dann, wenn ein vorheriger Auftrag abgeschlossen wird (Pull-Prinzip). Bei einer Simultanplanung für die Aufgaben 1 bis 4 wird hingegen bei Eintreffen eines neuen Auftrags ein neuer Gesamtplan erstellt, der die Zuordnung zu Kommissionierplätzen, die Zuordnung zu Regalen sowie die Sequenzierung der Transportaufträge beinhaltet. Das Ereignis „Kommissionierauftrag abgeschlossen“ löst dann keinen Planungsschritt aus.

Im Hinblick auf die Untersuchung solcher interdependenten Steuerungsverfahren ist die ereignisdiskrete Simulation der flexibelste Ansatz und bildet auch die Grundlage der vorliegenden Arbeit. Ein Bausteinkasten zur Simulation von RMFS muss dabei die Abbildung und Verwaltung unterschiedlichster Planungsprozesse zur Lösung einer beliebigen Kombination von Planungsaufgaben unterstützen. Der Nutzen der Simulation wurde bereits in früheren Arbeiten thematisiert. So beruhen die von Hazard et al. (2006) sowie von Merschformann et al. (2018b) konzipierten Experimentierumgebungen auf der ereignisdiskreten Simulation. Auch Lienert und Fottner (2017) setzen ereignisdiskrete Simulation zur Analyse von RMFS-Verfahren ein.

2.2 Verfügbare Software zur Simulation von RMFS

Hazard et al. (2006) entwickelten die Java-basierte Simulationsumgebung AlphabetSoup, die als Experimentierumgebung für Forschungsarbeiten konzipiert ist, sich aber aufgrund ihres hohen Abstraktionsgrads nicht für die Entwicklung realitätsnaher Modelle eignet. AlphabetSoup erreichte keine größere Verbreitung, und die Entwicklung wurde bereits im Jahr 2007 mit Version 0.92 eingestellt. Derselben Grundidee folgt die von Merschformann et al. (2018) in C# realisierte Simulationsumgebung RAWSim-O, die realitätsnähere Modelle erlaubt und vorrangig der Evaluation von agentenbasierten Lösungsansätzen dienen soll.

In industriellen Projekten zur Entwicklung von Logistikzentren werden heute meist kommerzielle Materialflusssimulatoren, wie z. B. Plant Simulation, eingesetzt. Derartige Simulatoren nutzen zur Modellierung von Unstetigförderern in der Regel ein Fahrzeug-und-Wegenetz-Konzept: Fahrzeuge bewegen sich auf Wegsegmenten, und die Wegeplanung erfolgt mittels des A*- oder des Floyd-Warshall-Algorithmus (Gutenschwager et al. 2012). Für die Abbildung von RMFS eignet sich dieser Ansatz nur bedingt, weil die Wegeplanungsverfahren nicht deadlocksicher sind und die Unterfahrbarkeit von Regalen durch unbeladene Roboter nicht berücksichtigen.

Zusätzlich zu weggebundenen Fahrzeugen bieten moderne Materialflusssimulatoren auch Elemente, die sich frei in der Fläche bewegen können. In Plant Simulation sind dies Werker und FTF. Allerdings sind beide Bausteintypen für die Abbildung von RMFS ungeeignet: Werker können zwar statische Hindernisse umgehen, erkennen aber Kollisionen mit anderen Werkern nicht und folgen auch nicht dem Raster eines RMFS-Layouts. Die in Plant Simulation 15 eingeführten flächenbeweglichen FTF benötigen zwar kein Wegenetz, dafür aber eine vorgegebene Sequenz von Wegpunkten, die ohne Routenplanung und Kollisionserkennung abgefahren werden. Somit fehlen Plant Simulation Standardbausteine, die für die Modellierung von mobilen Robotern in RMFS geeignet wären. Auch andere kommerzielle Simulatoren bieten gegenwärtig keine geeigneten Modellierungselemente an.

Lienert und Fottner (2017) schlagen einen Modellierungsansatz vor, der ein rasterförmiges Lagerlayout durch einen Graphen abbildet, dessen Knoten jeweils eine rechteckige Fläche repräsentieren, die einem Stellplatz oder einem Wegsegment entspricht. Das Konzept wurde bereits in Plant Simulation umgesetzt und ist sehr gut für die Abbildung von RMFS geeignet. Allerdings fokussieren die Arbeiten von Lienert und Fottner bisher auf Routing-Probleme.

3 Fachkonzept des Bausteinkastens

Jedes RMFS besteht aus einer Reihe von Basiselementen, die in einem Simulationsmodell abgebildet und verwaltet werden müssen. Während die Planungs- und Steuerungsverfahren eines RMFS sehr vielfältig sein können, bieten die Funktionen zur Verwaltung der RMFS-Elemente das Potential zur Standardisierung in einem Bausteinkasten. Benötigt werden Funktionen zur Verwaltung

- der Artikel mit ihren Beständen,
- der Elemente des Layouts,
- der Roboterflotte sowie
- der Kommissionier-, Transport- und Fahraufträge.

3.1 Elemente von RMFS

Zentrale Eingangsgröße jedes RMFS sind *Kommissionieraufträge*, die mehrere *Auftragspositionen* umfassen, wobei jede Auftragsposition das Kommissionieren einer bestimmten *Menge* genau eines *Artikels* verlangt. Der Bestand eines Artikels kann auf mehrere *Regale* verteilt sein. Umgekehrt enthält ein Regal meist mehrere Artikel. Für jeden Artikel und jedes Regal muss außer dem *aktuellen Bestand* auch die bereits *vorgemerkte Menge* verwaltet werden. Jeder Kommissionierauftrag wird im Zuge des Planungsprozesses einem *Kommissionierplatz* zugeordnet.

Zur Erfüllung der Kommissionieraufträge werden *Transportaufträge* gebildet. Ein Transportauftrag umfasst die Folge aller Transportschritte von der Fahrt zu einem Regalstellplatz, über das Aufnehmen des Regals, den Transport des Regals zu einem oder mehreren Kommissionierplätzen bis zum Absetzen des Regals. Eine Auftragsposition kann auf mehrere Regale – und damit auf mehrere Transportaufträge – verteilt werden. Um die Leistung eines RMFS zu steigern, können Positionen verschiedener Kommissionieraufträge in einem Transportauftrag zusammengefasst werden, sofern an den Kommissionierplätzen keine Termin- und Reihenfolge-restriktionen verletzt werden (Merschformann et al. 2018b).

Der Transportauftrag definiert, welche Kommissionierplätze in welcher Reihenfolge angefahren werden. Die *erwarteten Ankunftszeiten* an den Kommissionierplätzen können im Rahmen der Planung als Schätzwerte ermittelt werden, die möglicherweise von den später simulativ ermittelten *tatsächlichen Ankunftszeiten* abweichen.

Analog zum Ansatz von Lienert und Fottner (2017) wird das Layout des RMFS aus rechteckigen *Kacheln* gebildet. Die Navigationsmöglichkeiten zu benachbarten Kacheln werden explizit angegeben, so dass auch Einbahnstraßen und gesperrte Bereiche abbildbar sind. Den Kacheln werden spezielle *Rollen* zugeordnet, um sie als Regalstellplatz, Warteposition für einen Kommissionierplatz, Pickposition für einen Kommissionierplatz, Ladestation oder Default-Position für einen Roboter zu kennzeichnen. Diese Rollen müssen z. B. bei der Wegeplanung und der Auswahl von Stellplätzen für nicht mehr benötigte Regale berücksichtigt werden.

Jeder Transportauftrag wird von genau einem Roboter ausgeführt. Ein Transportauftrag erfordert in der Regel mehrere Transportschritte (Leerfahrt zum Regalstellplatz, Fahrt zum ersten Kommissionierplatz, Fahrten zu den folgenden Kommissionierplätzen, Fahrt zum Abstellplatz für das Regal), für die jeweils ein *Fahrauftrag* definiert wird. Jeder Fahrauftrag führt über eine Folge von Kacheln, deren Belegung zur Deadlock-Vermeidung vorgeplant wird, so dass für jede Kachel auf dem Weg eine *Reservierung* für ein definiertes *Zeitfenster* entsteht. Zudem kann der Roboter auf einer Kachel eine *Operation* ausführen, nämlich das Aufnehmen oder Abstellen eines Regals sowie den Übergang in einen Wartezustand (z. B. um das Ende eines Kommissionier- oder Batterieladevorgangs abzuwarten).

3.2 Planungsauslösende Ereignisse und Planungsprozesse

Die in Abschnitt 2.1 genannten Planungsaufgaben 1 bis 9 können wie bereits beschrieben sowohl separat als auch simultan gelöst werden, wobei hinsichtlich der Kombination der Planungsaufgaben und der Bearbeitungsreihenfolge verschiedene Varianten denkbar sind. Beispielsweise lassen sich die Planungsaufgaben 1 bis 4 in einem gemeinsamen Optimierungsmodell zusammenfassen, das heuristisch oder mittels eines geeigneten mathematischen Optimierungsverfahrens gelöst wird. Alternativ kann ein sequentieller Planungsprozess definiert werden, der die Teilaufgaben nacheinander und gegebenenfalls als Reaktion auf unterschiedliche Ereignisse zu unterschiedlichen Zeitpunkten behandelt.

Der in dieser Arbeit vorgestellte Bausteinkasten soll einerseits eine strukturierte Vorgehensweise zur Abbildung solcher Planungsprozessvarianten unterstützen und andererseits möglichst große Freiheiten bei der Problemlösung einräumen. Daher wurde folgender Ansatz gewählt: Für die Lagerverwaltung (Warehouse Management System, *WMS*) und für die Roboterwaltung (*RobotControl*) stellt jeweils ein

Baustein alle Funktionen und Datenstrukturen zum Zugriff auf Ressourcen, Artikel und Aufträge zur Verfügung. Die eigentlichen Planungsprozesse, im Sinne einer Strukturierung der einzelnen Planungsaufgaben, sind in Teilmodellen gekapselt. Grundlage für die Beschreibung der Planungsprozesse sind Modelle nach dem Business Process Model and Notation Standard (BPMN, vgl. Object Management Group 2013). Jede Aktivität im BPMN-Modell (jeder Planungsschritt) kann im Simulationsmodell durch eine separate Methode abgebildet werden.

Da verschiedene Planungsprozesse möglich sind, werden Prozessvarianten definiert. Abbildung 1 zeigt beispielhaft eine Variante eines Planungsprozesses als BPMN-Modell. Die Prozessvariante wird durch einen Ereignistyp ausgelöst (Abschluss eines Kommissioniervorgangs) und erfüllt die Planungsaufgaben 1 bis 4 (siehe Abschnitt 2.1) von der Sequenzierung der Kommissionieraufträge bis zur Zuordnung der Kommissionierauftragspositionen zu Regalen.

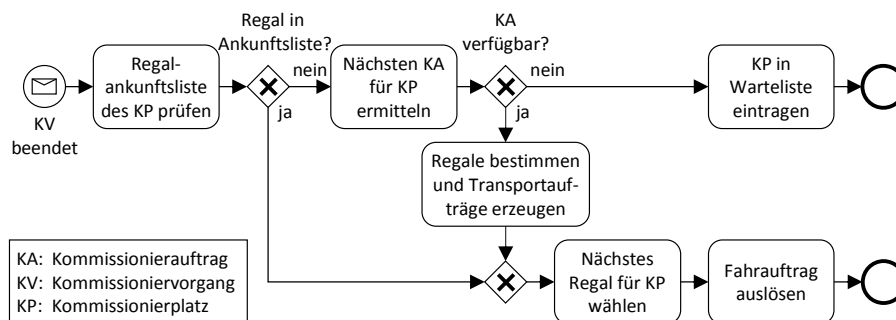


Abbildung 1: BPMN-Modell einer Planungsprozessvariante innerhalb des WMS

Um verschiedene Planungsprozesse in das Simulationsmodell einzubinden, muss auf relevante Ereignisse differenziert reagiert werden. Auf WMS-Ebene werden folgende potentiell planungsauslösende Ereignisse unterschieden:

- Eintreffen eines neuen Kommissionierauftrags
- Abschluss eines Transportauftrags
- Abschluss eines Kommissioniervorgangs (Regal bereit zur Weiterfahrt)
- Überschreiten eines Toleranzwertes für eine Plan-Ist-Abweichung
- Ende eines fest vorgegebenen Zeitintervalls für einen neuen Planungslauf

Der Eintritt eines dieser Ereignisse kann über die Ausführung einer entsprechenden Schnittstellenmethode einen Planungsprozess anstoßen. Dazu muss der Modellierer für jedes Ereignis bzw. jede Schnittstellenmethode spezifizieren, ob eine Verbindung zur jeweiligen Planungsprozessvariante existiert, und eine entsprechende, gegebenenfalls prozessspezifische Methode angeben. Abbildung 2 zeigt beispielhaft die unterschiedliche Einbindung von zwei Prozessvarianten auf der WMS-Ebene. Planungsprozessvariante 1 ist einem auslösenden Ereignis zugeordnet, Planungsprozessvariante 2 kann von zwei verschiedenen Ereignissen initiiert werden: Tritt ein planungsrelevantes Ereignis ein, wird zunächst die zugehörige Schnittstellenmethode aufgerufen. Diese initiiert den Planungsprozess, indem sie die zugeordnete Startmethode der jeweiligen Planungsprozessvariante aufruft, die wiederum weitere Methodenaufrufe zur Realisierung der einzelnen Planungsschritte steuert.

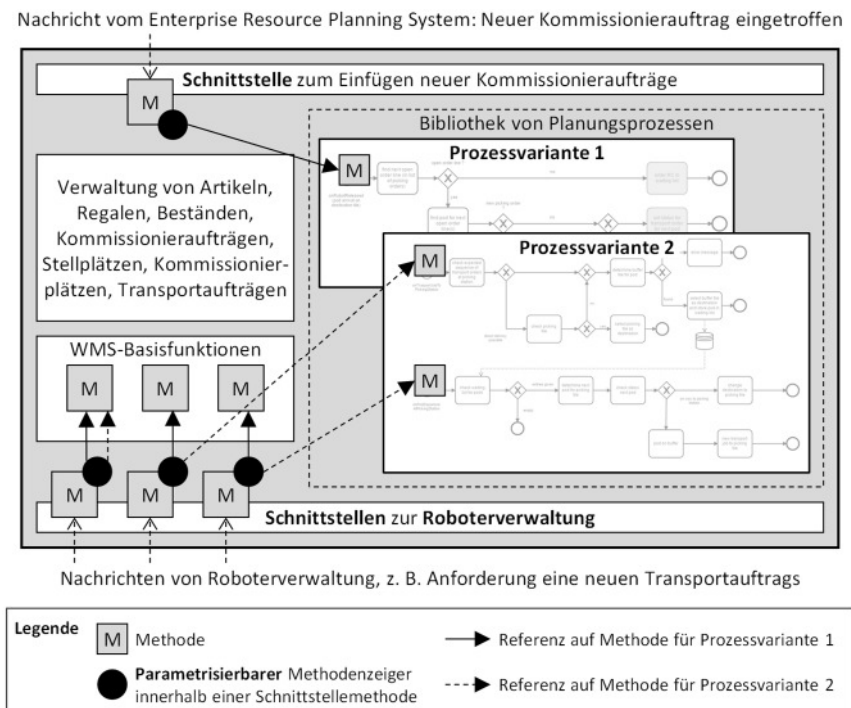


Abbildung 2: Einbindung konfigurierbarer Planungsprozesse

3.3 Masterdaten- und Layout-Generatoren

Um RMFS-Verfahren zu evaluieren, werden hinreichend große Testinstanzen benötigt. Ein Masterdaten-Generator kann genutzt werden, um die entsprechenden logistischen Daten (Artikelstammdaten, Belegung der Regale mit Beständen, Kommissionieraufträge mit Auftragspositionen) auf der Grundlage von Wahrscheinlichkeitsverteilungen zu erzeugen.

Der Layout-Generator dient dazu, über eine Tabelle die einzelnen Kacheln eines RMFS mit ihren Eigenschaften, Rollen sowie Nachbarschaftsbeziehungen zu definieren (siehe Abschnitt 3.1). Der Aufbau des Layouts erfolgt auf Basis dieser Informationen automatisiert. Der Modellierer kann das generierte Layout grafisch interaktiv nachbearbeiten, wobei die Änderungen auch in der zugrundeliegenden Tabelle zur Beschreibung des Layouts nachgezogen werden.

4 Implementierung in Plant Simulation

Das in Abschnitt 3 vorgestellte Konzept wurde als Baustein Kastens für das Simulationswerkzeug Plant Simulation prototypisch umgesetzt. Plant Simulation ist ein in der industriellen Praxis weitverbreitetes System und seine Offenheit erlaubt eine vergleichsweise einfache Umsetzung des vorgestellten Konzepts. Ein weiterer Vorteil liegt im Wegfindungsmechanismus (Gutenschwager et al. 2012). Vergleichbare andere Simulatoren nutzen den Floyd-Warshall-Algorithmus, um zu Beginn eines Simulationslaufs alle kürzesten Verbindungen zu berechnen. Diese

Logik ist für RMFS nicht anwendbar. Demgegenüber berechnet Plant Simulation die kürzeste Verbindung zwischen Start und Ziel während des Simulationslaufs jeweils neu für das aktuelle Wegenetz. Dieser Mechanismus kann durch geeignete Modellierung der Kacheln auch für RMFS genutzt werden (siehe unten).

Abbildung 3 zeigt ein RMFS-Modell und die zur Modellierung verwendeten Klassen. Die Ressourcen von RMFS werden mit Hilfe der Bausteine *Tiles* (Kacheln), *MobileRobot*, *Pod* (Regal) und *PickingStation* (Kommissionierplatz) abgebildet. Die Abbildung der übergeordneten Steuerlogik erfolgt mittels der Bausteine *WMS* und *RobotControl*, die gemäß dem vorgestellten Konzept als separate Klassen implementiert sind. Die Planungsprozessvarianten sind als Klassen im Ordner *Processes* abgelegt. In der Tabelle *t_Processes* wird festgelegt, welche Prozessvarianten im Simulationslauf genutzt werden sollen und wie diese Prozessvarianten konfiguriert sind, d. h. welche Ereignisse welche Methoden auslösen. Ein Nutzer des Bausteinkastens kann die Prozessvarianten-Klassen ableiten oder duplizieren, um Prozessvarianten zu modifizieren oder neue Prozessvarianten zu definieren. Der Modellierer wählt für Lager- und Roboterverwaltung jeweils eine Prozessvariante pro Planungsprozess aus. Diese wird dann automatisch in den jeweiligen Baustein integriert.

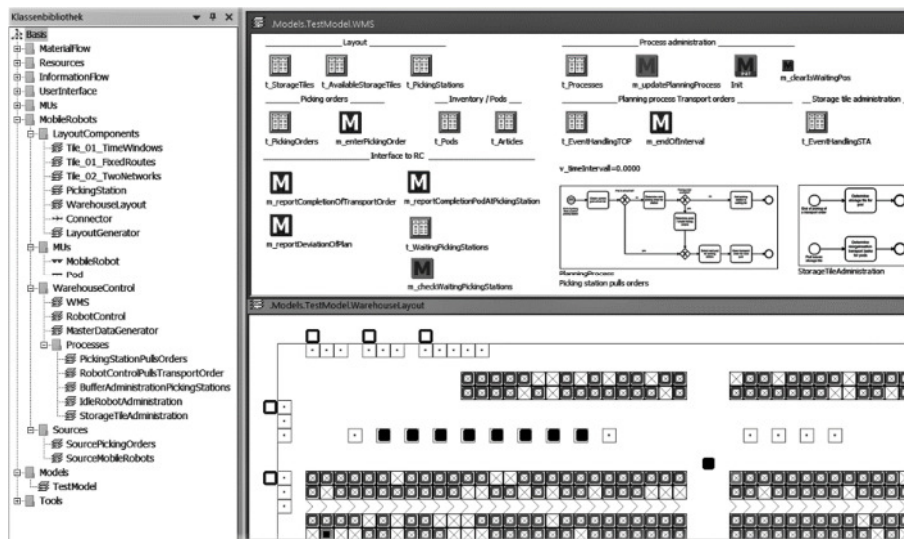


Abbildung 3: Beispielmodell

Auf Materialflussebene werden Kacheln (*Tiles*) mit Hilfe von Netzwerk-Bausteinen abgebildet, die durch Kanten miteinander verknüpft sind. Die interne Logik setzt die verschiedenen Rollen der Kacheln um (z. B. Pickposition für Kommissionierplatz, Regalstellplatz). Für die Wegeplanung stehen drei alternative Verfahren zur Verfügung, die auf unterschiedlichen Realisierungen der Tiles beruhen:

- Definition fester „Wege“ zwischen einzelnen Zielen des Layouts: Für jeden Weg gibt es eine vorgegebene Reihe zu durchlaufender Kacheln.

- Nutzung der Basisfunktionalität von Plant Simulation zum Finden kürzester Wege in Kombination mit einer dynamischen Deadlock-Erkennung auf Ebene der aktuellen Belegung benachbarter Kacheln: Hierbei sind in jeder Kachel zwei Realisierungen der internen Track-Bausteine und Schnittstellen zu benachbarten Kacheln umgesetzt, die somit zwei unterschiedliche Wegenetze im Gesamtsystem definieren. Ein Netz dient dem Routing von unbeladenen Robotern, das zweite dem von beladenen Robotern, die im Unterschied zu unbeladenen Robotern Regale nicht unterfahren können.
- Anwendung eines Routing-Verfahrens unter Berücksichtigung von Zeitfenstern für jede Kachel, wobei die Zeitfensterlänge von dem Zeitbedarf für das Durchfahren einer Kachel abhängt (Lienert und Fottner 2017).

Der Baustein *PickingStation* bildet einen Kommissionierplatz generisch ab. Die Verwaltung von Wartepositionen an Kommissionierplätzen wird in einem eigenen Planungsprozess abgebildet, der auch das Nachziehen von Regalen Richtung Kommissionierplatz als Teilprozess beinhaltet. Auch für diesen Prozess (*Buffer-AdministrationPickingStations*), der in die Klasse *RobotControl* eingebunden ist, können Varianten erzeugt werden.

Die einzelnen Bausteinfunctionalitäten wurden anhand unterschiedlicher Modelle validiert. Der Fokus lag dabei auf einer geeigneten Wegefindung einschließlich der Deadlock-Vermeidung. Der Ansatz, Kacheln über Zeitfenster für Fahrten zu reservieren, hat sich dabei unter Inkaufnahme des höheren Rechenaufwandes als überaus stabil erwiesen.

5 Zusammenfassung und Ausblick

Aktuellen Materialflusssimulatoren mangelt es an Standardbausteinen zur Modellierung von RMFS und damit an Funktionalität zur Evaluation von RMFS-Verfahren. Der Beitrag stellt einen Bausteinkasten für das Simulationssystem Plant Simulation vor, der diese Lücke schließen soll. Gegenüber vergleichbaren Lösungen zeichnet sich das vorgestellte Konzept insbesondere durch eine explizite Abbildung der Varianten von Planungsprozessen aus, die während der Simulation durch Ereignisse ausgelöst werden. Damit eignet sich der Bausteinkasten nicht nur für die Untersuchung von agentenbasierten Steuerungsverfahren, sondern auch für die Analyse zentraler Steuerungsansätze, die auf Regelsystemen oder Optimierungsverfahren basieren.

In Folgearbeiten wird der Bausteinkasten zunächst funktional erweitert (z. B. um die Behandlung von Nachschubaufträgen). Anschließend soll er bei der Evaluation und dem Vergleich verschiedener RMFS-Verfahren zum Einsatz kommen. Ob das Ziel einer für den Modellierer effizienten Arbeitsumgebung erreicht wird, kann erst beurteilt werden, wenn der Bausteinkasten auch von verschiedenen Personen zur Evaluation neuer, bisher nicht umgesetzter RMFS-Verfahren eingesetzt wird.

Literatur

Boysen, N.; Briskorn, D.; Emde, S.: Parts-to-picker based order processing in a rack-moving mobile robots environment. *European Journal of Operational Research* 262 (2017) 2, S. 550-562.

- Emde, S.; Boysen, N.: Kommissionierung mithilfe mobiler Lagerroboter – Ablauf und Planungsaufgaben. In: Glock, C.; Grosse, E. (Hrsg.): Warehousing 4.0: Technische Lösungen und Managementkonzepte für die Lagerlogistik der Zukunft. Lauda-Königshofen: B+G Wissenschaftsverlag 2017, S. 103-113.
- Enright, J. J.; Wurman, P. R.: Optimization and Coordinated Autonomy in Mobile In: Automated Action Planning for Autonomous Mobile Robots: Papers from the 2011 AAI Workshop, San Francisco, 7. August 2011, S. 33-38.
- Gutenschwager, K.; Radtke, A.; Völker, S.; Zeller, G.: The Shortest Path: Comparison of Different Approaches and Implementations for the Automatic Routing of Vehicles. In: Laroque, C.; Himmelspach, J.; Pasupathy, R.; Rose, O.; Uhrmacher, A. M. (Hrsg.): Proceedings of the 2012 Winter Simulation Conference (WSC), Berlin, 10.-12. December 2012, S. 3312-3323.
- Hanson, R.; Medbo, L.; Johansson, M. I.: Performance Characteristics of Robotic Mobile Fulfillment Systems in Order Picking Applications. In: IFAC PapersOnLine 51 (2018) 11, S. 1493–1498.
- Hazard, C. J.; Wurman, P. R.; D’Andrea, R.: Alphabet Soup: A Tested for Studying Resource Allocation in Multi-Vehicle Systems. In: Proceedings of the 2006 AAI Workshop on Auction Mechanisms for Robot Coordination, Boston, 17. July 2006, S. 23-30.
- Lamballais, T.; Roy, D.; De Koster, M. B. M.: Estimating performance in a Robotic Mobile Fulfillment System. European Journal of Operational Research 256 (2017) 3, S. 976-990.
- Lienert, T.; Fottner, J.: Entwicklung einer generischen Simulationsmethode für das zeitfensterbasierte Routing Fahrerloser Transportfahrzeuge. In: Tagungsband des 13. Fachkolloquiums der WGTL, Graz, 20.-21. September 2017, S. 307-322.
- Merschformann, M.; Xie, L.; Erdmann, D. (2018a) Multi-Agent Path Finding with Kinematic Constraints for Robotic Mobile Fulfillment Systems. arxiv.org (2018).
- Merschformann, M.; Xie, L.; Li, H. (2018b) RAWSim-O: A Simulation Framework for Robotic Mobile Fulfillment Systems. Logistics Research 11 (2018) 8, S. 1-11.
- Nobbe, C. G.: Vergleich technischer Implementierungen für GridFlow-Systeme. Karlsruhe: KIT Scientific Publishing 2015.
- Object Management Group: Business Process Model and Notation (BPMN), Version 2.0.2: 2013.
- Wulfrat, M.: Is Kiva Systems a Good Fit for Your Distribution Center? – An Unbiased Distribution Consultant Evaluation. MWPVL International Inc, 2012, http://www.mwpvl.com/html/kiva_systems.html.
- Wurman, P. R.; D’Andrea, R.; Mountz, M.: Coordinating Hundreds of Cooperative, Autonomous Vehicles in Warehouses. In: Cheetham, W.; Goker, M. (Hrsg.): IAAI’07 Proceedings of the 19th National Conference on Innovative Applications of Artificial Intelligence - Volume 2. Menlo Park: AAAI Press 2007, S. 1752-1759.
- Yuan, Z.; Gong, Y.: Improving the Speed of Delivery for Robotic Warehouses. IFAC PapersOnLine 49 (2016) 12, S. 1164-1168.
- Yuan, Z.; Gong, Y.: Bot-In-Time Delivery for Robotic Mobile Fulfillment Systems. IEEE Transactions on Engineering Management 64 (2017) 1, S. 83-93.