

Dezentrale Koordination und Kooperation im Internet der Dinge

Dipl.-Inf. R. Chisu, Dipl.-Ing. F. Kuzmany, Prof. W. A. Günthner

Technische Universität München

Lehrstuhl für Fördertechnik Materialfluss Logistik fml

Das Internet der Dinge als neuartiges Modell für eine dezentrale und hierarchielose Steuerung innerbetrieblicher Materialflusssysteme beginnt, in der Welt der Logistik Fuß zu fassen. Dieser Beitrag befasst sich dabei mit zwei technischen Fragestellungen, ohne deren Beantwortung ein industrieller Einsatz dieser Steuerungsarchitektur aber als wenig sinnvoll erscheint. Im ersten Teil wird gezeigt, wie ein Baukasten aus standardisierten, funktionsorientierten Modulen für die Umsetzung kunden- und projektspezifischer Abläufe herangezogen werden kann: Transporteinheiten werden dabei zu den Trägern der Geschäftslogik eines Systems und arbeiten logistische Prozesse ab, die als Folge einzelner Funktionen modelliert werden. Der zweite Teil adressiert komplexe und für logistische Anlagen typische Materialflussstrategien. Anhand konkreter Beispiele wird aufgezeigt, wie autonome Entitäten bzw. Fördertechnikmodule in der Lage sind, durch relativ einfache und szenariounabhängige Kostenfunktionen und Verhaltensregeln auch komplexe Strategien dezentral umzusetzen.

Einleitung

Das Umfeld produzierender Unternehmen hat sich in den letzten Jahren stark verändert. Viele Unternehmen konzentrieren sich zunehmend auf ihre Kernkompetenzen, beigeordnete Bereiche oder solche mit niedrigem Wertschöpfungsgrad werden ausgegliedert. Zusammen mit der Globalisierung der Absatz- und Beschaffungsmärkte hat dies zu einer viel stärkeren Vernetzung von Unternehmen und einer Zunahme des Transportaufkommens geführt. Dies macht die Organisation logistischer und produktionstechnischer Prozesse zu einer immer komplexeren Aufgabe, Prognosen und feste Annahmen werden zunehmend unrealistisch. Dieser Effekt wird durch Entwicklungen auf der innerbetrieblichen Ebene weiter verstärkt. Um in übersättigten Märkten bestehen zu können, müssen Unternehmen ihre Produktpalette immer weiter diversifizieren – die Anzahl angebotener Varianten steigt, Innovations- und Produktlebenszyklen werden kürzer. Individualisierte Produkte stellen dabei die extremste Form variantenreicher Produktion dar: Der Kunde gestaltet sein Produkt selbst und bestellt im Regelfall ein einziges Exemplar. Die marktorientierte Fertigung wird somit immer mehr von der auftragsorientierten Fertigung abgelöst – im immer häufiger werdenden Extremfall in Losgröße eins.

Besonders betroffen ist hierbei die Logistik. Diese muss zunehmend komplexe Abläufe und Materialflüsse steuern, flexibel und hochdynamisch reagieren und dabei möglichst geringe Kosten verursachen. Vor allem im innerbetrieblichen Bereich hat es die Technik aber nicht geschafft, mit den sich schnell verändernden Anforderungen Schritt zu halten. Hochautomatisierte Materialflusssysteme bieten zwar Effizienz- und Kostenvorteile, sind aber in ihrer heutigen Ausführung sehr starr und unflexibel gegenüber Änderungen. Dies liegt größtenteils an der zentralen und projektspezifisch ausgelegten Steuerungsarchitektur, die zudem für größere Anlagen eine kaum durchschaubare Komplexität erreichen kann. Zentrale Steuerungen sind sehr weit verbreitet, aber angesichts der immer wichtiger werdenden Flexibilität und Wandlungsfähigkeit nicht mehr zeitgemäß. Zu groß ist der finanzielle und zeitliche Aufwand für die Programmierung des zentralen Materialflussrechners, für die Integration und Inbetriebnahme der verschiedenen Subsysteme und Bereichssteuerungen und vor allem für den Umbau.

Das Internet der Dinge

Als Alternative zu herkömmlichen Steuerungsarchitekturen werden schon seit einigen Jahren dezentrale Systeme untersucht oder schon eingesetzt [Beu-06, Gün-00, Hom-05, May-09, Rit-03, Sau-00]. Dieses Konzept beruht auf der Gliederung der Gesamtaufgabe der Materialflussteuerung in kleinere, überschaubare Teilprobleme und ihrer Verteilung auf möglichst standardisierte Funktionskomponenten bzw. Module. Diese kapseln eine bestimmte logistische Funktion [Wil-06] und können beliebig miteinander kombiniert werden, um so kooperativ den Materialfluss zu steuern. Materialflussanlagen verhalten sich dem entsprechend immer mehr wie ein Datennetzwerk, in dem autonom agierende Switches und Router Datenpakete selbständig weiterleiten und sich kontinuierlich über Veränderungen der Topologie oder Auslastungen von Verbindung austauschen. In einem „Internet der Dinge“ werden Logik bzw. Steuerungskompetenzen aus den verschiedenen Hierarchieebenen herkömmlicher zentraler Architekturen zusammengefasst und auf autonom agierende Materialflussmodule und Transporteinheiten verteilt [Gün-08a] (siehe Abbildung 1).

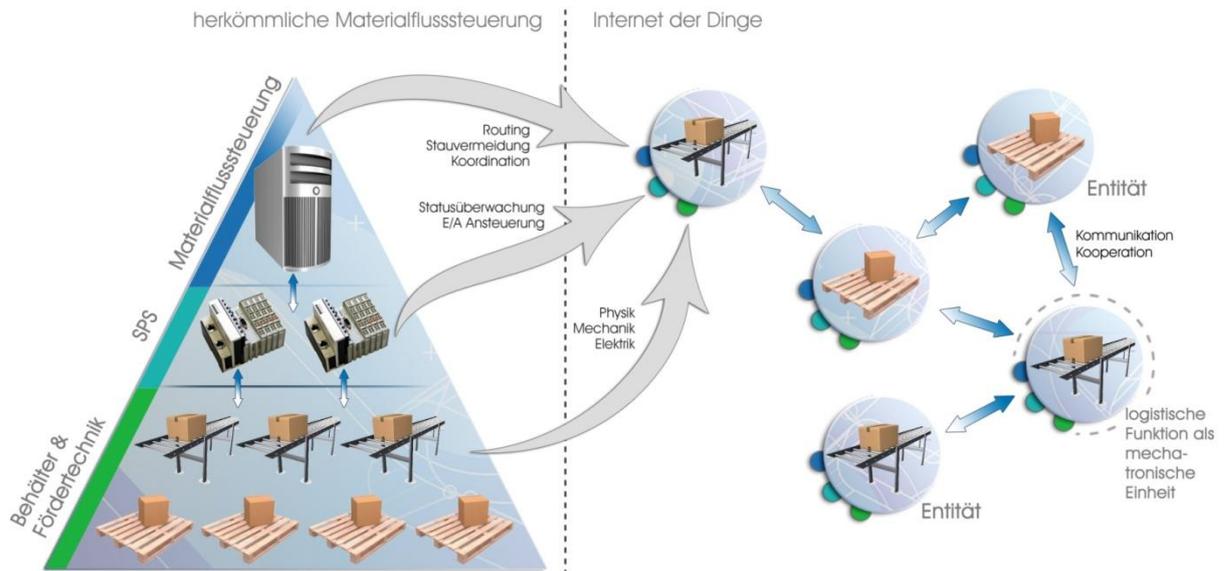


Abbildung 1 Das Internet der Dinge ist eine dezentrale und hierarchielose Materialflusssteuerung

Darüber hinaus können auch Softwareprogramme, so genannte „Dienste“, zum Einsatz kommen. Diese haben im Regelfall keinen Einfluss auf die Materialflusssteuerung an sich, sind aber notwendig, um Aufgaben wie Visualisierung, Backups oder die Verwaltung von Verzeichnissen umzusetzen oder um Datenschnittstellen zu externen Systemen zu realisieren. Die elementare Einheit dieser kooperativen Materialflusssteuerung wird dabei als Entität bezeichnet und kann in den genannten drei Ausprägungen – Modul, Transporteinheit oder Dienst – vorkommen (siehe Abbildung 2). Jede Entität wird mit einer eigenen Programmlogik – beispielsweise einem Softwareagenten – ausgestattet, die für die Verwaltung und Abarbeitung aller Funktionen der jeweiligen Entität sorgt. Damit greifen auch Transporteinheiten – anders als passive Datenpakete oder E-Mails im Internet – aktiv in die Materialflusssteuerung ein. Sie suchen sich selbst ihr Ziel, wählen bei mehreren Alternativen selbständig die günstigste davon aus und veranlassen und überwachen ihren Transport. Die Fördertechnikmodule hingegen können als Dienstleister aufgefasst werden, die die von den Transporteinheiten erteilten Aufträge erfüllen. Dabei setzen die Module die eigentliche Materialflusssteuerung um: sie berechnen Routen, berücksichtigen Förderstrategien und führen Optimierungen durch.

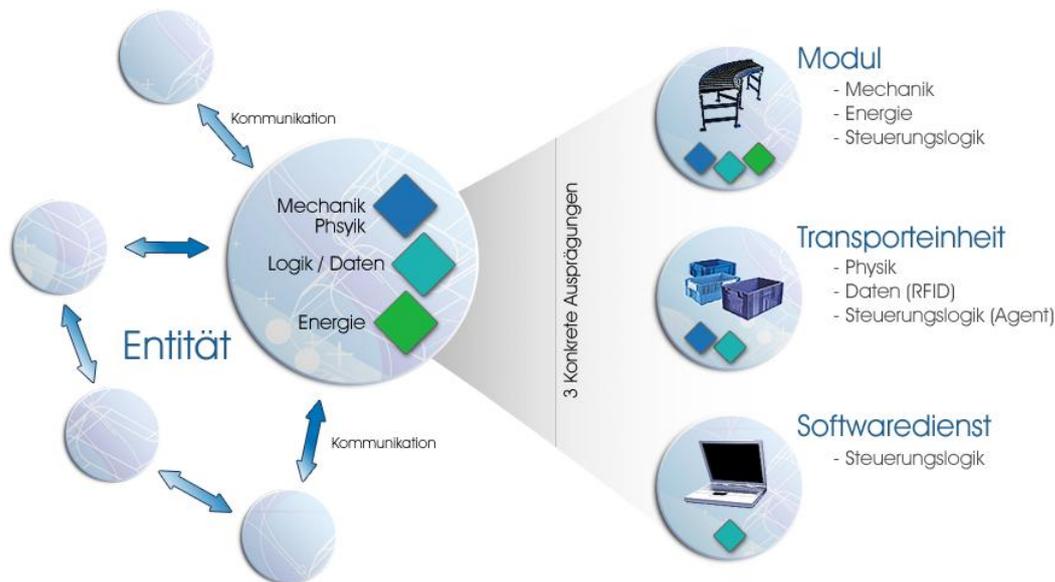


Abbildung 2 Entitäten: Der grundlegende Baustein im Internet der Dinge

Entitäten sind vollständig autonom und interagieren miteinander nur über standardisierte Kommunikationsprotokolle und –inhalte. Diese Kapselung erlaubt die Behandlung von Modulen, Transporteinheiten und Diensten als „Black Box“ – ihre interne Gestaltung und die lokal verwendeten Softwarealgorithmen und Datenstrukturen sind anderen Entitäten unbekannt und für das Gesamtsystem irrelevant. Dies ermöglicht die einfache Austauschbarkeit von Entitäten, was gerade im Fall der (Fördertechnik-)Module von großem Vorteil ist, und erleichtert somit die Integration.

Die Leistungsfähigkeit des Internet der Dinge wurde simulativ bereits für sehr große Materialflussanlagen nachgewiesen [Roi-07], Demonstrationsanlagen haben auch die technische Machbarkeit dieses Modells gezeigt [Wil-06]. Bisherige Arbeiten betrachten aber größtenteils nur die Auftragsdisposition und die Wegplanung. Diese sind zwar sehr grundlegende Funktionen, genügen in den meisten Fällen jedoch nicht, um die teils sehr komplexen intralogistischen Prozesse abzudecken. Dieser Beitrag adressiert daher zwei bisher weitgehend offen gebliebene Fragestellungen dezentraler Materialflusssteuerung:

1. Die Realisierbarkeit nicht standardisierbarer, anlagenspezifischer Abläufe mittels eines Baukastens aus standardisierten Modulen
2. Die dezentrale Implementierung von Materialflussstrategien über das einfache Routing hinaus, z.B. für das Sortieren von Transporteinheiten oder die Ressourcenallokation

Umsetzung projektspezifischer Abläufe

Der Einsatz eines Baukastens aus Standardkomponenten bietet zwar viele technische und auch wirtschaftliche Vorteile [Gün-08b], wirft aber gleichzeitig die Frage auf, ob bzw. wie die von System zu System stark unterschiedliche Geschäftslogik einer Anlage implementiert werden kann, ohne dass dabei die Umprogrammierung aller Module notwendig wird. Um das Internet der Dinge industriell einsetzbar zu machen, ist also neben den Aspekten der Materialflusssteuerung an sich auch die einfache Realisierbarkeit anlagenspezifischer Prozesse von großer Relevanz.

Die Module des Internet der Dinge werden anhand standardisierbarer logistischer Grundfunktionen, beispielsweise Fördern, Verteilen oder Lagern [Arn-95], identifiziert und gestaltet. Für den Anlagenbetreiber ist aber letztlich nur die (nicht standardisierbare) Funktion des Gesamtsystems, die sich aus der zielgerichteten Kombination und Verkettung einzelner Grundfunktionen ergibt, von Bedeutung. Als Beispiel für eine solche Geschäftslogik soll hierbei ein (vereinfachter) Arbeitsablauf für ein Gepäckstück am Flughafen dienen:

1. Ein eingetragener Koffer muss zuerst in einer Röntgenanlage gescannt werden.
2. Besteht Verdacht auf einen gefährlichen Inhalt des Gepäckstücks, wird dieses in einer zweiten Röntgenanlage mit höherer Genauigkeit überprüft.
3. Zum wiederholten Mal als unsicher eingestufte Koffer werden ausgeschleust und manuell weiterbearbeitet. Im ersten oder zweiten Schritt für sicher befundene Gepäckstücke werden zum Verladen freigegeben.
4. Ist das korrekte Flugzeug für den Verladevorgang geöffnet, kann das Gepäck sofort zum Ziel transportiert werden.
5. Ist dies nicht der Fall, muss der Koffer in einem Frühgepäckspeicher zwischengelagert werden.
6. Sobald der Flug verladen werden kann, muss der Koffer den Frühgepäckspeicher verlassen und zum Ziel befördert werden.

Dieser Prozess kann als Sequenz einzelner Funktionen aufgefasst werden, die jeweils von einzelnen Modulen erbracht werden können – beispielsweise Durchleuchten oder Zwischenlagern. Um solche Abläufe flexibel modellieren und dezentral implementieren zu können, sind zwei Dinge notwendig:

- die projektspezifische Geschäftslogik muss von der standardisierten Steuerungslogik der Module getrennt werden,
- es muss eine Methode zur aufwandsminimalen Definition und Veränderung von Arbeitsabläufen bzw. Workflows gefunden werden.

Transporteinheiten als Träger der Geschäftslogik

Die Geschäftslogik eines Systems bezieht sich auf Transporteinheiten und gibt an, was wann und unter welchen Umständen mit dem Stückgut zu geschehen hat. Die Materialflusssteuerung hingegen hat die Aufgabe, die dafür notwendigen Transporte umzusetzen und dabei optimale Durchlaufzeiten und Durchsätze zu erreichen. So wie die Materialflusssteuerung auf Fördertechnikmodule verteilt wird, wird im Internet der Dinge die Umsetzung der Geschäftslogik den Transporteinheiten zugewiesen.

Transporteinheiten werden zu Anfang ihrer Bearbeitung – beispielsweise am Check-In, am Wareneingang oder bei der Auslagerung – mit der Abarbeitung eines entsprechenden Arbeitsablaufs bzw. Workflows beauftragt. Dieser besteht aus einer Sequenz einzelner funktionaler Schritte („Durchleuchten“, „Zwischenlagern“, „Verladen“ ...), wobei auch Verzweigungen („Wenn Koffer sicher, verladen, sonst ausschleusen.“) und Prioritäten („Wenn möglich, verladen, sonst zwischenlagern.“) vorkommen können.

Jede Transporteinheit erhält einen eigenen, individuellen Workflow – es ist also durchaus möglich, verschiedenen TEs unterschiedliche Arbeitsabläufe zu erteilen, was beispielsweise in einem Kommissionierlager für die Umsetzung verschiedener Kommissionierstrategien notwendig sein kann. Die interne Abarbeitung eines solchen Arbeitsablaufs, also das Durchlaufen der Sequenz und das bedingte Entscheiden über den nächsten auszuführenden Schritt, wie auch die notwendige Kommunikation mit Modulen sind durch einfache, universell einsetzbare Mechanismen zu implementieren. Der Softwareagent einer Transporteinheit benötigt zum einen eine so genannte Workflow-Engine, die in der Lage ist, beliebige Arbeitsabläufe zu durchlaufen (siehe auch nächsten Abschnitt „Modellierung von Workflows“). Zum anderen muss er das Gesamtsystem nach Modulen durchsuchen können, die eine bestimmte Funktion – z.B. „Durchleuchten“ – erbringen, und aus den eventuell mehreren Alternativen die günstigste auswählen. Das UML-Sequenzdiagramm in Abbildung 3 zeigt die dabei notwendige Kommunikation zwischen Transporteinheit und anderen Entitäten.

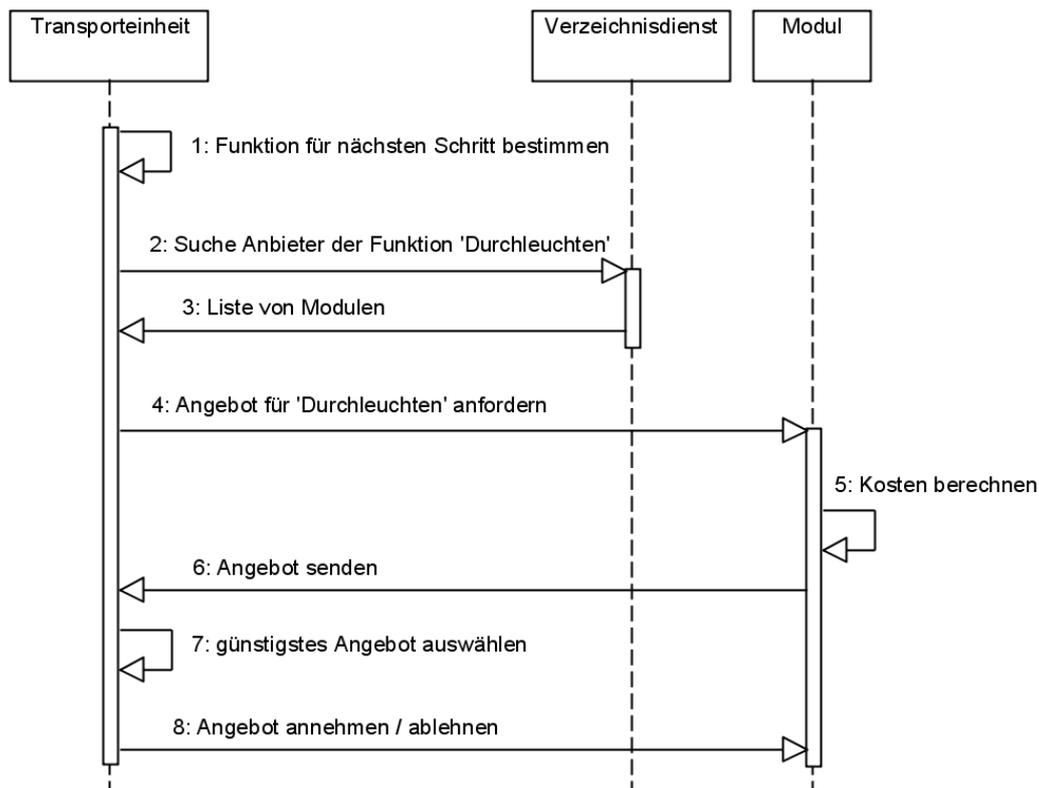


Abbildung 3 Abarbeiten eines Workflows durch eine Transporteinheit

Modellierung von Workflows

Workflows müssen wie auch in zentralen Systemen vom Anlagenprogrammierer oder –bediener festgelegt werden, da es einem technischen System unmöglich ist, aus sich selbst heraus seine eigentliche Funktion und die dafür notwendigen Abläufe zu erkennen. Eine Modellierungssprache für Arbeitsabläufe muss also sowohl maschinenlesbar bzw. für Transporteinheiten automatisiert interpretierbar als auch für einen Bediener intuitiv verwendbar sein. Eine geeignete Methode für die Beschreibung von Workflows sind daher so genannte High-Level Petri Netze. Petri Netze sind ein graphisch darstellbares und formal abgesichertes mathematisches Modell zur Darstellung von Systemtransitionen [Abe-90, Bau-96]. Petri Netze sind bipartite Graphen mit folgenden Elementen (siehe auch Abbildung 4):

- **Stellen** entsprechen den Zuständen eines Systems und werden als Kreise dargestellt. Die in einem Zustand des Gesamtsystems aktiven Stellen werden mit einer oder mehreren Marken gekennzeichnet.
- **Transitionen** repräsentieren einen Zustandsübergang im System und werden als Vierecke dargestellt.
- **Kanten** verbinden eine Stelle mit einer Transition, jedoch nie zwei Transitionen oder zwei Stellen. Sie werden als gerichtete Pfeile dargestellt.

- **Marken** werden als Punkte innerhalb von Stellen dargestellt und kennzeichnen die gerade aktiven Zustände im System. Marken sind nicht unterscheidbar.

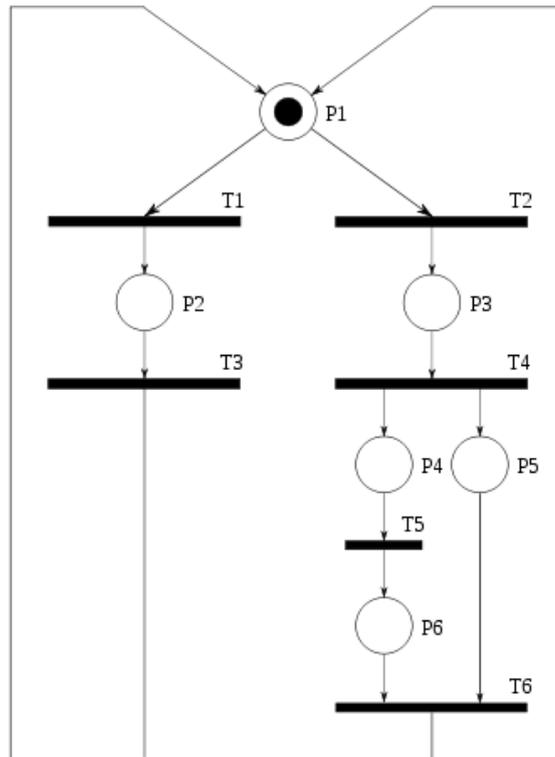


Abbildung 4 Einfaches Petri Netz

Als Erweiterung zur klassischen Version definieren High-Level Petri Netze zusätzliche Mechanismen, die für die Definition deterministischer intralogistischer Workflows notwendig sind. Besonders wichtig sind hierbei:

- Die Priorisierung von Transitionen bzw. Zustandsübergängen („Verladen ist wichtiger als Zwischenlagern“),
- Das Zuweisen von „Farben“ bzw. unterschiedlichen Eigenschaften zu den Marken, die bei Zustandsübergängen geprüft werden [Jen-91, Jen-92] („Wenn Koffer unsicher, ausschleusen“).

Obwohl Petri Netze vor allem für die Analyse nebenläufiger Systeme herangezogen werden und somit mehrere Marken bzw. mehrere aktive Systemzustände gleichzeitig existieren können, wird für die Modellierung eines Workflows eine einzige Marke benötigt. Diese gibt den aktuellen bzw. nächsten Schritt an, den eine Transporteinheit zu bearbeiten hat. Abbildung 5 stellt den eingangs beschriebenen Workflow als High-Level Petri Netz dar. Die im Systemzustand bzw. Workflowschritt „Durchleuchten 1“ befindliche Marke gibt an, was mit dem Koffer geschehen muss. Die Transporteinheit muss also ein Modul suchen, welches diese Funktion (zu möglichst günstigen Kosten) anbietet und sich dann dort hin transportieren lassen. Nach Abschluss der

Sicherheitsprüfung wird, in Abhängigkeit von deren Ergebnis, der nächste Schritt bestimmt und auf dieselbe Art bearbeitet.

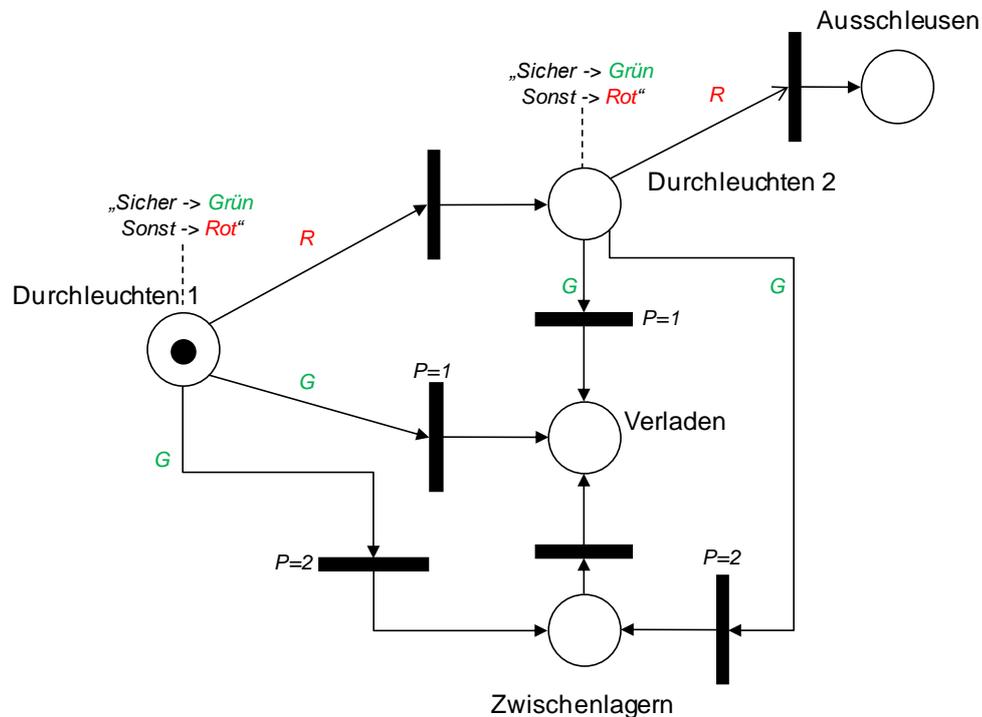


Abbildung 5 High-Level Petri Netz für einen Workflow zur Abarbeitung eines Gepäckstücks am Flughafen

Dezentrale Materialflusstategien

Workflows wurden als Sequenz von Funktionen, die jeweils von einem einzigen Modul erbracht werden, definiert. Dabei sind aber unter Umständen Strategien zu beachten, die zwar der Transporteinheit unbekannt, für die Materialflusststeuerung aber von Bedeutung sind. So stellt sich bezüglich des voran gegangenen Flughafenbeispiels die Frage, wie der Schritt „Zwischenlagern“ tatsächlich durchgeführt wird, bzw. welche Randbedingungen und Lagerstrategien dabei berücksichtigt werden müssen. Eine der dabei wichtigsten Vorgänge ist das Sortieren von Transporteinheiten nach verschiedenen Kriterien – z.B. nach ihrer Flugnummer oder Abflugzeit.

Ein anderes Beispiel dafür, dass die Erbringung einer Funktion nicht immer durch rein lokale Entscheidungen und Aktionen möglich ist, ist der Transport. Um bspw. eine Palette zu einem von ihr gewählten Ziel zu befördern, müssen sich mehrere Module in ihrem Handeln abstimmen. Interessant ist hierbei vor allem der Fall, in dem mehrere Module um eine Ressource konkurrieren, beispielsweise wenn mehrere Querverschiebewägen (oder Regalbediengeräte) auf derselben Schiene fahren.

Die nächsten Abschnitte greifen diese Herausforderungen auf und zeigen Möglichkeiten zur dezentralen Realisierung einer Sorterverwaltung und der Ressourcenallokation bzw. Koordination mehrere Module.

Sortieren

Das Sortieren von Transporteinheiten ist immer dann notwendig, wenn Stückgut für die Vereinfachung oder Beschleunigung nachfolgender Prozesse in eine bestimmte Ordnung gebracht werden muss (siehe Abbildung 6). Dabei werden die Sortierkriterien je nach Anwendung unterschiedlich festgelegt und beziehen sich beispielsweise auf das Ziel, den Inhalt, die Priorität oder sonstige Attribute einer Transporteinheit.

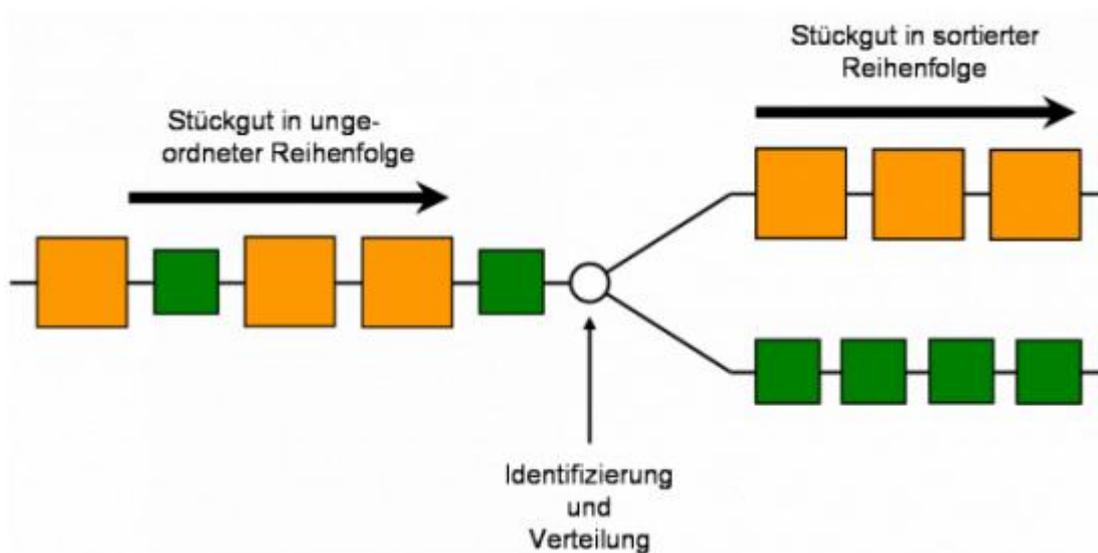


Abbildung 6 Sortieren von Stückgut

Im beschriebenen Flughafenszenario steht für das Zwischenlagern ein so genannter Frühgepäckspeicher zur Verfügung. Dieser besteht oftmals aus mehreren Pufferbahnen mit unterschiedlichen Rollen:

- einige Sorterbahnen werden je einem Flug zugeordnet und nehmen nur für das entsprechende Flugzeug bestimmte Gepäck auf,
- andere Bahnen werden einem Zeitfenster zugeordnet und nehmen Gepäckstücke auf, die in dem jeweiligen Zeitraum verladen werden, unabhängig von deren Flug.

Gemäß der funktionsorientierten Modularisierung wird nun jede einzelne Bahn als eigenständiges und selbststeuerndes Modul aufgefasst. Die Sortierlogik muss also auf die einzelnen Sorterbahnen so verteilt werden, dass jede Bahn für sich und mit keinem oder nur minimalem Wissen über den Zustand der anderen Bahnen entscheiden kann, ob sie einen Koffer aufnimmt bzw. zu welchen Kosten. Zur Berechnung dieser Kosten wertet jede Sorterbahn unabhängig von allen anderen

Modulen eine Kostenfunktion aus, wobei der Flug und der Verladezeitpunkt des Koffers die Grundlage für die Berechnung bilden. Diese Kostenfunktion kann sehr einfach gestaltet werden und bildet dezentral die gewünschte Sortierlogik ab (siehe Formel 1):

- Kosten = 0, wenn die Bahn bereits für den Flug reserviert ist, in den die Transporteinheit verladen werden muss,
- Kosten = 1, wenn die Bahn einem Zeitfenster zugeordnet ist, in dem der Koffer voraussichtlich verladen wird,
- Kosten = 2, wenn die Bahn noch frei ist, den Koffer aber prinzipiell aufnehmen darf,
- Kosten = ∞ in allen anderen Fällen – wenn die Bahn z.B. voll ist oder für einen anderen Flug reserviert ist.

$$k(TE) = \begin{cases} 0, & \text{wenn } b \triangleq z(TE) \text{ und Bahn nicht voll} \\ 1, & \text{wenn } b \triangleq t(TE) \text{ und Bahn nicht voll} \\ 2, & \text{wenn } b = \emptyset \text{ und Bahn reservierbar für } z(TE) \text{ oder } t(TE) \\ \infty, & \text{sonst} \end{cases}$$

Formel 1 Kostenfunktion für einen einfachen Sorter

Dabei sind:

- b die aktuelle Belegung der die Funktion berechnenden Bahn (für eine freie Bahn sei $b = \emptyset$)
- $z(TE)$ das Ziel der Transporteinheit TE bzw. der Flug, in den der Koffer verladen werden muss
- $t(TE)$ das Abflugzeitfenster des Gepäckstücks TE

Darüber hinaus ist es ebenso möglich, den Belegungsgrad einer Bahn in die Kostenfunktion mit einfließen zu lassen. Sollen beispielsweise mehrere Bahnen mit identischem Sortierkriterium gleichmäßig befüllt werden, ist es möglich, die aktuelle Auslastung der Bahn im Nachkommabereich der bereits beschriebenen Kosten darzustellen: Eine zu 10% volle Bahn berechnet beispielsweise ein Angebot von 1,1, eine zu 25% volle ein Angebot 1,25. Auf diese Weise wird eine Transporteinheit immer die am wenigsten ausgelastete und gemäß der Sortierlogik optimale Bahn anfahren.

Sortieren mit Paarbildung

Eine vor allem in der Lagerlogistik anzutreffende komplexere Variante des Sortierproblems ist ein Sorter mit Paarbildung. Bei diesem gilt die zusätzliche Anforderung, dass zwischen Transporteinheiten, die nebeneinander befindlichen Bahnen zugeordnet werden, ein gewisser Zusammenhang bestehen muss. Abbildung 7 zeigt ein solches Beispiel: die auf sechs Sorterbahnen verteilten Transporteinheiten werden von einem Doppelquerverschiebewagen abgeholt und anschließend über weitere Förderer zu insgesamt vier AKLs weitertransportiert.

Da ein optimaler Durchsatz nur dann erreicht werden kann, wenn möglichst immer ein doppelter Lastwechsel zwischen Sorter und Querverschiebewagen und zwischen Querverschiebewagen und den anschließenden Förderern durchgeführt werden kann, müssen die Transporteinheiten so vorsortiert werden, dass sie in Bezug auf ihr Transportziel (AKL1 bis AKL4) paarweise auf benachbarten Sorterbahnen liegen. In diesem konkreten Beispiel muss also eine Transporteinheit mit Ziel AKL1 immer links (in Förderrichtung) neben einer Transporteinheit mit Ziel AKL2 einsortiert werden, während eine Transporteinheit mit Ziel AKL3 immer rechts von einer Einheit mit Ziel AKL4 liegen muss.

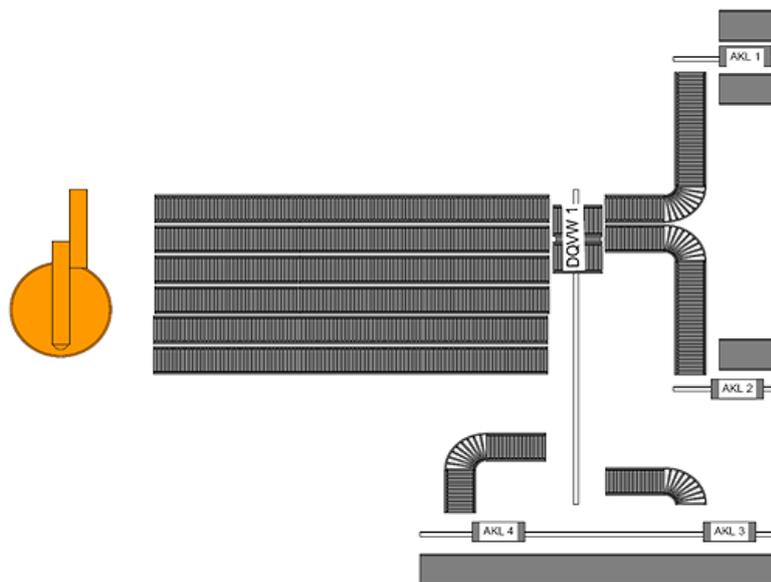


Abbildung 7 Sorter mit Paarbildung

Auch in diesem Fall lässt sich die Sorterverwaltung mittels einer relativ einfachen Kostenfunktion dezentral umsetzen. Dabei zu beachten ist aber, dass die Sorterbahnen in diesem Fall über weitere Daten verfügen müssen, und zwar

- die zu realisierenden Paare,
- die Namen ihrer direkten Nachbarbahnen (soweit vorhanden),
- die aktuelle Belegung der direkten Nachbarbahnen.

Während die ersten zwei Informationen statisch sind und sich nur mit einem Umbau der Anlage ändern, verändert sich der dritte Punkt recht häufig während des Betriebs. Jede Bahn muss also, vor Berechnung der Kostenfunktion, seine direkten Nachbarn über deren aktuelle Belegung befragen oder diese Information beispielsweise über eine zentrale Datenaustauschplattform beziehen.

Die Kostenfunktion gestaltet sich dann wie folgt:

$$k(TE) = \begin{cases} 0, & \text{wenn } b \cong z(TE) \text{ und } b_n \cong c(z(TE)) \\ 1, & \text{wenn } b = \emptyset \text{ und } b_n \cong c(z(TE)) \\ 2, & \text{wenn } b \cong z(TE) \text{ und } b_n = \emptyset \\ 3, & \text{wenn } b = \emptyset \text{ und } b_n \neq \emptyset \\ 4, & \text{wenn } b \cong z(TE) \text{ und } b_n \neq c(z(TE)) \text{ und } b_n \neq \emptyset, \text{ ODER} \\ & \text{wenn } b \neq z(TE) \text{ und } b \neq \emptyset \text{ und } b_n \cong c(z(TE)), \text{ ODER} \\ & \text{wenn } b = \emptyset \text{ und } c(b_n) \neq \emptyset \text{ und } c(b_n) \neq z(TE) \\ 5, & \text{wenn } b \neq \emptyset \text{ und } b \neq z(TE) \text{ und } b_n \neq \emptyset \text{ und } b_n \neq c(z(TE)) \\ \infty, & \text{wenn Bahn } i \text{ voll} \end{cases}$$

Formel 2 Kostenfunktion für einen Sorter mit Paarbildung

Dabei sind:

- b die aktuelle Belegung der die Funktion berechnenden Bahn (für eine freie Bahn sei $b = \emptyset$)
- b_n die Belegung einer Nachbarbahn
- $z(TE)$ das Ziel der Transporteinheit TE
- $c(z(TE))$ das Ziel, welches eine Transporteinheit haben muss, um mit TE ein Paar zu bilden

Diese Funktion gewährleistet die korrekte Befüllung des Sorters mit den gewünschten Paaren (Kosten 0, 1 und 2), ermöglicht aber gleichzeitig das Ausnutzen der gesamten Sorterkapazität: Ist die Realisierung der gewünschten Paare nicht möglich, weil die dafür notwendigen Bahnen voll sind, wird eine Transporteinheit auf eine andere, noch nicht volle Bahn umgeleitet (Kosten 4 und 5). Durch diese Vernachlässigung der Paarbildungsvorschrift wird zwar der Durchsatz am Punkt des Doppelverschiebewagens vom Optimum abweichen, andererseits wird das Entstehen eines Staus am Verteilelement vor dem Sorter verhindert. Leere Bahnen mit leeren Nachbarbahnen – was dem Initialisierungszustand des Systems entspricht – sind mit Kosten = 3 eine schlechtere Option als Bahnen, bei denen eine korrekte Paarbildung möglich ist, aber immernoch dem Zerstören der Paarbildungsvorschrift vorzuziehen. Bereits voll besetzte Bahnen werden wegen der unendlichen Kosten von einer Transporteinheit niemals als Ziel ausgewählt.

Sequenzbildung

Die Bildung von Sequenzen von Transporteinheiten – wie sie beispielsweise bei der Kommissionierung oder beim Versand notwendig ist – kann als Sonderfall des Sortierens aufgefasst werden. Der Unterschied besteht darin, dass das „Sortierkriterium“ dynamisch ist und sich mit Aufnahme einer Transporteinheit jedes Mal ändert.

Für die Kostenfunktion genügen im einfachsten Fall zwei Werte: 0, wenn eine Transporteinheit die nächste Position in der gewünschten (und von einem externen System, z.B. dem LVS oder einem Kommissionierer vorgegebenen) Sequenz darstellt und unendlich in allen anderen Fällen. Auch hier sind Verfeinerungen

möglich, beispielsweise um Prioritäten verschiedener Kommissionieraufträge abzubilden oder um zu unterscheiden, ob die genaue Sequenz der Transporteinheiten eingehalten werden muss, oder ob die Anlieferung der gewünschten Güter in korrekter Anzahl schon genügt.

Dezentrales Ressourcenmanagement

An vielen Stellen in einem Materialflusssystem ist es notwendig, die Aufteilung von Ressourcen zwischen verschiedenen Systemteilnehmern zu organisieren. Beispiele hierfür sind Zusammenführungen des Materialflusses oder von mehreren Regalbediengeräten, Elektrohängebahnkatzen oder Querverschiebewägen gemeinsam genutzte Fahrwege. Eine Grundvoraussetzung für den störungsfreien Betrieb ist hier die Verhinderung von Blockadesituationen, die in der Regel dadurch aufgelöst werden, dass ein Ressourcennutzer einem anderen Vorrang gewähren muss. Die Vorfahrt wird dabei durch den Nutzern zugeordnete Prioritäten geregelt, welche meist noch durch übergeordnete Ziele beeinflusst sind. Dies kann z.B. die Verfolgung einer LIFO- bzw. FIFO-Strategie bzw. die bevorzugte Abarbeitung von Eilaufträgen sein.

In herkömmlichen Materialflusssystemen werden diese Vorgaben durch einen Bereichs- oder Materialflussrechner umgesetzt, der gegenüber dezentral gesteuerten Systemen den Vorteil hat, jederzeit auf ein konsistentes Datenmodell zugreifen und wie ein „Absolutherrscher“ die ihm unterstellten Fördertechnikelemente befehligen zu können. Dies kann jedoch, wie eingangs beschrieben, zu komplexen und schwer überschaubaren Steuerungsprogrammen führen, welche nur schwer erweiterbar sind. Im Folgenden soll am Beispiel mehrerer, auf einer gemeinsamen Schiene fahrenden Querverschiebewägen gezeigt werden, dass auch in diesem Fall eine dezentrale und auf relativ einfachen Regeln aufbauende Steuerung möglich und erstrebenswert ist.

Ein Querverschiebewagen (QVW) hat die Aufgabe, Transporteinheiten (TE) von einer oder mehreren Ankunftstrecken auf eine oder mehrere Abtransportstrecken zu befördern. Dazu bewegt er sich auf einer Schiene senkrecht zu den Zuführ-/Abtransportstrecken und verfügt selbst über einen Förderer, welcher die Lastübergabe durchführt.

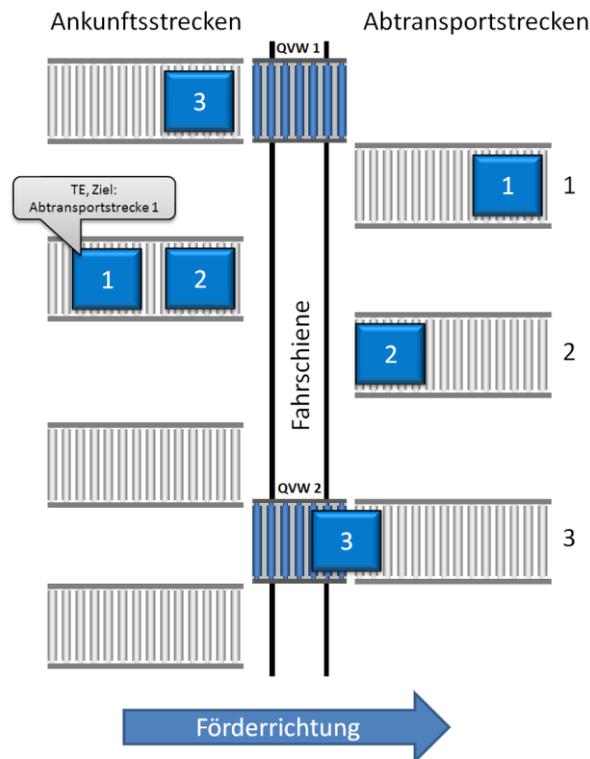


Abbildung 8 Querverschiebewägen

Liegt ein Transportauftrag vor, so sei das Arbeitspiel eines QVW wie folgt:

1. Fahrt zur Ankunftsstrecke
2. Lastaufnahme
3. Fahrt zur Abtransportstrecke
4. Lastabgabe

Im konkreten Beispiel soll davon ausgegangen werden, dass die ankommenden Transporteinheiten nach dem FIFO Prinzip weitergefördert werden. Warten also auf verschiedenen Ankunftsstrecken TEs auf ihren Transport, so ist diejenige bevorzugt zu befördern, welche am längsten gewartet hat. Zusätzlich soll immer derjenige freie QVW den Auftrag übernehmen, welcher den kürzesten Anfahrtsweg hat, um so die Transportwege und damit die Transportzeit zu minimieren. Diese Vorgaben führen zu einer ersten Prioritätsregelung:

$$P(TE, QVW) = t_W(TE) - t_{Anfahrt}(QVW)$$

Die Auftragsdisposition läuft damit wie folgt ab:

Jede Transporteinheit überwacht ihre Wartezeit t_W und meldet diese an die Querverschiebewägen. Diese berechnen ihre Anfahrtszeit $t_{Anfahrt}$ und berechnen $P(TE, QVW)$, welche wiederum an die Transporteinheit übermittelt wird. Auf diese Weise verfügen zum einen die Transporteinheiten über eine priorisierte Liste von

möglichen Transporteuren und zum anderen die Querverschiebewägen über eine Liste von bevorzugten TE. Der QVW kann dann unter den TE, welche sein Gebot angenommen haben, die für ihn günstigste auswählen. Nun wird die TE zum Transport reserviert und der QVW beginnt mit der Zielanfahrt.

Während dieser Zielanfahrt kann es bereits zu konkurrierenden Zielen der QVW und dadurch zu Blockadesituationen kommen. Ab jetzt gilt:

$$P_{\text{Zielanfahrt}}(\text{QVW}) = t_W(\text{TE}) + c * t_{\text{QVW}}(\text{TE}) - \rho(\text{QVW}) + R$$

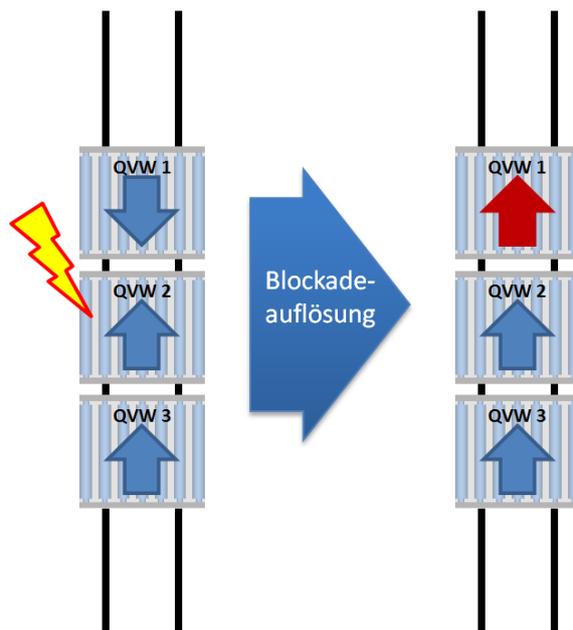
Hierbei fließt neben der Wartezeit $t_W(\text{TE})$ zusätzlich mit Gewichtung $c > 1$ die Zeit ein, welche eine TE bereits auf dem QVW verbringt $t_{\text{QVW}}(\text{TE})$. Dies führt dazu, dass ein beladener QVW immer eine höhere Priorität besitzt als ein unbeladener, so dass ein bereits begonnener Auftrag beendet werden kann, ohne dass ein freier QVW die Ausführung behindert. $\rho(\text{QVW})$ beschreibt die Auslastung eines QVW und sorgt in der Prioritätsgleichung dafür, dass bei einem Patt immer der QVW zum Zuge kommt, welcher bisher die geringere Auslastung hatte, was eine gleichmäßige Auslastung der QVW garantiert. Um den unwahrscheinlichen Fall zu verhindern, dass trotz Allem zwei QVW die gleiche Priorität erhalten, geht schließlich eine Zufallszahl R in die Berechnung mit ein. Diese sollte so klein gewählt werden, dass sie zwar eine Pattsituation verhindert, die anderen Faktoren aber nicht überragt.

Tritt nun ein Zielkonflikt auf, so verhält sich jeder QVW nach folgenden Regeln:

- Falls der blockierende QVW eine höhere Priorität hat, fahre auch in dessen Richtung
- Wenn du einem Blockierer ausweichst, addiere die Priorität des Blockierers zu deiner eigenen, d.h.:

$$P_{\text{Blockade}}(\text{QVW}) = P(\text{Blockierer}) + P_{\text{Zielanfahrt}}(\text{QVW})$$

Die zweite Regel führt in größeren Systemen dazu, dass die Priorität quasi von einem Fahrzeug zum nächsten „vererbt“ wird und auf diese Weise eine gewichtete Mehrheitsentscheidung bzgl. der Fahrtrichtung getroffen wird. Ein Beispiel dafür ist in Abbildung 9 zu sehen. Zwar wäre QVW1 in der Lage, QVW2 zu verschieben. Dazu müsste aber zusätzlich QVW3 seine geplante Fahrtrichtung ändern. Die beiden QVW 2 und 3 sind aber gemeinsam „stark genug“, um QVW 1 zu überstimmen, so dass dieser ausweichen muss.



Wobei:

$$P_{\text{Zielanfahrt}}(\text{QVW3}) > P_{\text{Zielanfahrt}}(\text{QVW1}) > P_{\text{Zielanfahrt}}(\text{QVW2})$$

$$P_{\text{Zielanfahrt}}(\text{QVW1}) < P_{\text{Zielanfahrt}}(\text{QVW2}) + P_{\text{Zielanfahrt}}(\text{QVW3}) = P_{\text{Blockade}}(\text{QVW1})$$

Abbildung 9 Blockadeauflösung

Sobald die Blockade beendet ist bzw. ein Blockierer seinen Auftrag erledigt hat, sinkt dessen Priorität schlagartig, da er nun unbeladen ist. Nun kann der bis dato unterlegene QVW zum Zuge kommen.

Die bisher beschriebenen Prioritäts- und Verhaltensregeln führen zwar schon zu einem blockadefreien Ablauf, jedoch keineswegs zu einem hohen Durchsatz, da ein Konflikt erst aufgelöst wird, wenn die QVW schon auf Kollisionskurs sind. Es ist daher möglich, das bisherige Verhalten noch zu optimieren. Dazu sind verschiedene zusätzliche Verhaltensregeln denkbar, welche sowohl einzeln als auch in Kombination einsetzbar sind:

1. Auftrag bei Blockade stornieren

Falls du auf dem Weg zur Ankunftsstrecke von einem anderen QVW blockiert wirst, suche dir, falls möglich, einen neuen Auftrag, dessen Startpunkt in Fahrtrichtung des Blockierers liegt. Diese Regel verhindert, dass die Ziele der einzelnen QVW über Kreuz liegen, so dass sie sich nicht unnötig gegenseitig behindern.

2. Anfahrtsweg reservieren

Reserviere den Weg zur Ankunftsstrecke im Voraus, so dass andere QVW diesen nur noch in der gleichen Fahrtrichtung befahren dürfen. Diese Regel verfolgt die Idee, Blockaden nicht aufzulösen, sondern sie bereits im Vorfeld zu verhindern.

3. Zielfahrt reservieren

Reserviere den Weg von der Ankunftsstrecke bis zum Zielpunkt im Voraus, so dass andere QVW diesen nur noch in der gleichen Fahrtrichtung befahren dürfen.

4. Schätze Auftragsbearbeitungszeit

Die eingangs beschriebene Auftragsdisposition bezog ausschließlich freie QVW mit ein. Diejenigen, welche gerade mit der Abarbeitung eines Auftrags beschäftigt sind, sind daran noch nicht beteiligt. Daher kann die Berechnung der Priorität zur Auftragsdisposition $P(TE, QVW)$ so erweitert werden, dass sich auch belegte QVW an der Auftragsvergabe beteiligen können. Die Berechnung der Priorität wird dann wie folgt erweitert:

$$P(TE, QVW) = t_W(TE) - t_{\text{Anfahrt}}(QVW) - t_{\text{Auftragsbearbeitung}}(QVW)$$

Wobei $t_{\text{Auftragsbearbeitung}}$ der Zeit entspricht, die voraussichtlich verstreicht bis der QVW wieder zur Verfügung steht. Je nachdem, in welcher Phase seines Arbeitsspiels sich dieser befindet, gehen dabei Lastübergabezeiten und/oder noch verbleibende Fahrzeiten in die Berechnung mit ein.

Im Rahmen einer Simulation wurde die Leistungsverbesserung durch diese Strategien anhand eines Systems, das über drei QVW mit 10 Ankunftsstrecken und 8 Abtransportstrecken verfügt, überprüft. Dazu wurde der Durchsatz beim Betrieb ohne Verhaltensregeln mit demjenigen, der durch Anwendung einer zusätzlichen Verhaltensregel erzielt wird, verglichen (siehe Abbildung 10).

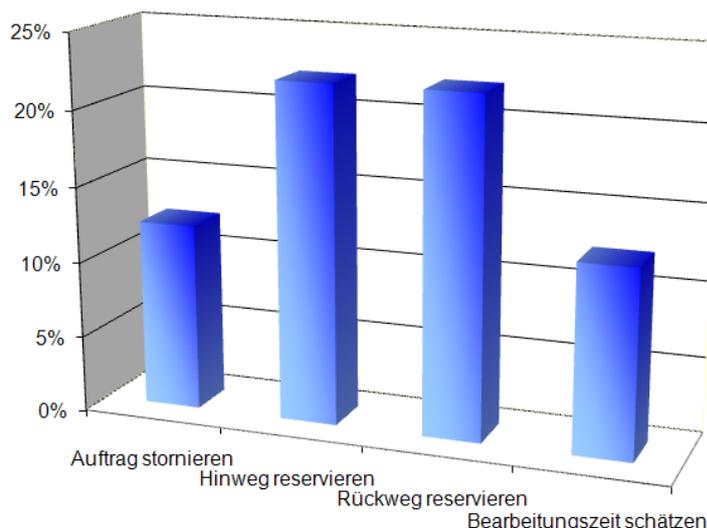


Abbildung 10 Durchsatzsteigerung durch Anwendung von Verhaltensregeln (einzeln)

Dabei zeigt sich, dass die Verhaltensregeln 2 und 3 jeweils die größten Auswirkungen auf den Durchsatz haben. Hier bestätigt sich die Vermutung, dass diejenigen Regeln, die Blockaden schon im Vorfeld verhindern, am wirkungsvollsten sind.

Anschließend wurde eine kombinierte Anwendung der Verhaltensregeln untersucht. So bewirkt die Kombination von Regel 1 & 2 bereits eine Durchsatzsteigerung von 40%. Alle Regeln zusammen erreichen sogar 54% (siehe Abbildung 11).

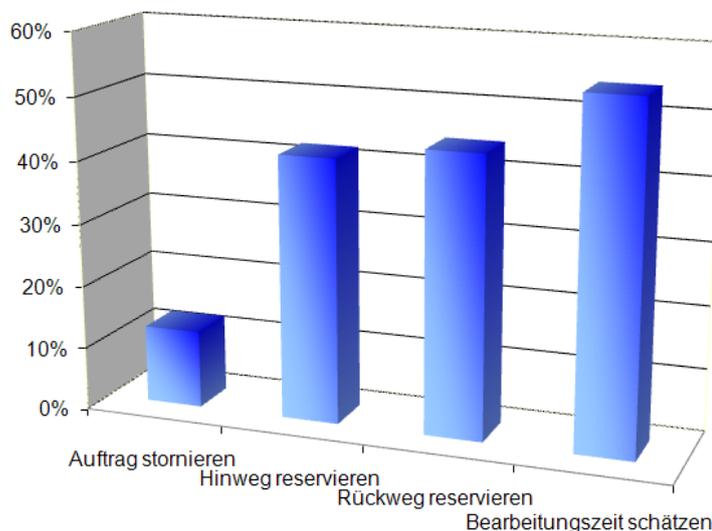


Abbildung 11 Durchsatzsteigerung durch Anwendung von Verhaltensregeln (kombiniert)

Die Auslastung der QVW ist in diesem System, das unter Überlast betrieben wurde, (theoretisch unendlicher Stau auf den Ankunftsstrecken) bei ca. 76%. Dieser im Vergleich zu einem einzeln betriebenen QVW niedrige Auslastungsgrad ist darauf zurückzuführen, dass sich Zielkonflikte und dadurch bedingte Behinderungen nicht gänzlich vermeiden lassen, wenn drei Fahrzeuge um einen Schienenweg konkurrieren.

Zusammenfassend lässt sich sagen, dass sich auch in einem dezentral gesteuerten System mit relativ einfachen Mitteln komplexe Verhaltensweisen umsetzen lassen, die den Vergleich mit zentraler Steuerung nicht scheuen müssen. Darüber hinaus haben aber dezentral gesteuerte Systeme den Vorteil einer übersichtlicheren Programmierung und der problemlosen Erweiterbarkeit. Die vorgestellten Regeln sind unabhängig von der Anzahl der QVW im System und der Ankunfts- bzw. Abtransportstrecken. Die Implementierung der Steuerungslogik ist damit nicht nur auf unterschiedliche Einsatzszenarien von Querverschiebewägen übertragbar, sondern ist auch für die Ansteuerung von Regalbediengeräten oder Elektrohängebahnkatzen einsetzbar.

Zusammenfassung

Materialflusssysteme müssen immer höheren Anforderungen gerecht werden. Dazu zählen vor allem die Flexibilität bzw. Wandelbarkeit und die Dynamik. Das Internet

der Dinge als eine neuartige, dezentrale und hierarchielose Materialflusssteuerung verspricht hier große Vorteile gegenüber heutigen Systemen: Anlagen können aus standardisierten und selbststeuernden Fördertechnikmodulen aufgebaut werden, intelligente Transporteinheiten beteiligen sich aktiv an der Steuerung logistischer Prozesse. Zentrale Instanzen wie der Materialflussrechner entfallen, projektspezifische Programmieraufwände werden stark verringert, die Inbetriebnahme wird vereinfacht.

Dieser Artikel hat gezeigt, wie es eine getrennte Betrachtung der Geschäftslogik und der Materialflusssteuerung erlaubt, beliebige logistische Abläufe mit standardisierten Modulen umzusetzen. Für die Modellierung beliebiger logistischer Workflows können beispielsweise High-Level Petri Netze herangezogen werden. Mit einer „Workflow-Engine“ ausgestattete Transporteinheiten sind dann in der Lage, die so beschriebenen Prozesse abzuarbeiten und gewährleisten auf diese Weise das zielgerichtete Zusammenspiel der autonomen Einheiten im Internet der Dinge.

Darüber hinaus wurde gezeigt, wie auch komplexe Materialflussstrategien dezentral umgesetzt werden können. Koordinations- und Kooperationsmuster für Module ergeben sich dabei aus einfachen lokalen Entscheidungsregeln und Kostenfunktionen, Transporteinheiten benötigen nur einfache und universell einsetzbare Service-Discovery Mechanismen für das Auffinden von Zielen und das Durchführen von Auktionen.

Literaturverzeichnis

- [Abe-90] Abel, D.: Petri-Netze für Ingenieure – Modellbildung und Analyse diskret gesteuerter Systeme, Berlin: Springer Verlag, 1990, ISBN 3-540-51814-2
- [Arn-95] Arnold, D.: Materialflußlehre, Vieweg, 1995, ISBN 3-528-03033-X
- [Bau-96] Baumgarten, B.: Petri-Netze - Grundlagen und Anwendungen, Heidelberg: Spektrum Akademischer Verlag, 1996, ISBN 3-8274-0175-5
- [Beu-06] Das BEUMER autover®-System – ein aktives DCV System, Prospekt der Beumer Maschinenfabrik GmbH & Co. KG Deutschland, http://www.beumer.com/htcms/get_file.php4?pid=346&ref=pdf&lang=de (Abgerufen am 11.11.2008)
- [Gün-00] Günthner W.A.; Reinhart, G. (Hrsg.): Abschlussbericht MATVAR - Materialflusssysteme für variable Fertigungssegmente im dynamischen Produktionsumfeld, Herbert Utz Verlag Wissenschaft, München 2000.

- [Gün-08a] Günthner, W.A.; Chisu, R.; Kuzmany, F.: Internet der Dinge - Steuern ohne Hierarchie, IN: F+H Fördern und Heben, Ausgabe September 2008, S. 494-497
- [Gün-08b] Günthner, W.A.; Chisu, R.; Kuzmany, F.: Internet der Dinge – Zukunftstechnologie mit Kostenvorteil, IN: F+H Fördern und Heben, Ausgabe Oktober 2008, S. 556-558
- [Hom-05] ten Hompel, M.; Liekenbrock, D.; Stuer, P.: Realtime Logistics: echtzeitnahe Steuerung von Materialflusssystemen auf Basis autonomer Agenten und Entitäten, IN: Logistics Journal: Nicht referierte Veröffentlichungen, 2005, ISSN 1860-5923
- [Jen-91] Jensen, K.; Rosenberg, G.. High-level Petri-Nets: Theory and Application. Springer Verlag, Berlin. 1991. ISBN 038754125X.
- [Jen-92] Jensen, K.: Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Springer Verlag, Berlin. 1992. ISBN 0387555978.
- [May-09] Mayer, S. H.: Development of a completely decentralized control system for continuous conveyors, Dissertation, Universität Karlsruhe (TH), April 2009
- [Rit-03] Ritter, A.: Ein Multi-Agenten-System für mobile Einrichtungen in Produktionssystemen, Diss., Universität Stuttgart, Jost Jetter Verlag Heimsheim, ISBN 3-936947-07-4, 2003
- [Roi-07] Roidl, M.; Follert, G.: Simulation von multiagentenbasierten Materialflusssteuerungen; Beiträge der 37. Jahrestagung der Gesellschaft für Informatik e.V.; Bremen 2007
- [Sau-00] Sauer, J.; Freese, T.; Teschke, T.: Towards agent-based multi-site scheduling; In: Jürgen Sauer, Jana Köhler (Hrsg.): Proc. of the ECAI 2000 Workshop on the New Results in Planning, Scheduling, and Design, S.123-130, 2000
- [Wil-06] Wilke, M.: Wandelbare automatisierte Materialflusssysteme für dynamische Produktionsstrukturen, Dissertation, TU München, 2006

Das Forschungs- und Entwicklungsprojekt „Internet der Dinge“, in dessen Rahmen die hier vorgestellten Konzepte entstanden sind, wird mit Mitteln des Bundesministeriums für Bildung und Forschung (BMBF) innerhalb des

Rahmenkonzeptes „Forschung für die Produktion von morgen“ gefördert und vom Projektträger Forschungszentrum Karlsruhe (PTKA) betreut.