

Prof. Dr.-Ing. Dipl.-Wi.-Ing. Willibald A. Günthner
Dipl.-Inf. Razvan Chisu
Dipl.-Ing. Florian Kuzmany

Internet der Dinge

-

Steuern ohne Hierarchie



Juni 2008

Das „Internet der Dinge“ ist die Antwort auf die Herausforderungen der modernen Intralogistik. Das intelligente, bedarfsorientierte Zusammenspiel autonomer Einheiten ersetzt zentral gesteuerte und unflexible Prozesse. Eine stark verringerte Systemkomplexität sowie höhere Robustheit, Wandelbarkeit und Erweiterbarkeit sind die Folge. Die organisatorischen und technologischen Konzepte wurden bereits entwickelt; das Internet der Dinge wird Wirklichkeit. In diesem Artikel bieten die Autoren einen Einblick in einen der zentralen Aspekte: die Softwarearchitektur des Internets der Dinge.

Internet der Dinge - Steuern ohne Hierarchie

Im voran gegangenen Artikel wurde das Modularisierungskonzept für das Internet der Dinge vorgestellt: Ein Netzwerk aus intelligenten Fördertechnikmodulen, Transporteinheiten und Softwarediensten steuert den Materialfluss auf kooperative Weise, ohne dass zentrale Steuerungsinstanzen wie z.B. ein Materialflussrechner notwendig sind.

Im vorliegenden Text wird nun die softwaretechnische Umsetzung dieses Konzeptes beleuchtet. Um dezentral gesteuerte Materialflusssysteme umzusetzen, werden die aus der Informatik bekannten Multiagentensysteme herangezogen und mit klassischen SPS-Programmen kombiniert. Auf diese Weise lassen sich sowohl komplexe Funktionen wie Wegplanung, Koordination und Optimierung als auch echtzeitkritische Aufgaben wie E/A-Verarbeitung auf einer einheitlichen und kostengünstigen Hardwareplattform realisieren und mit minimalem Aufwand für verschiedene Aufgabenstellungen wieder verwenden. Darüber hinaus verdient vor Allem die Kommunikation in einem echtzeitfähigen Materialflusssystem zwischen autonomen Einheiten besondere Aufmerksamkeit. Diesem Thema widmet sich der letzte Teil des Artikels.

Materialfluss ohne Materialflussrechner

Baukastensysteme und verteilte Rechenkapazität, beispielsweise in Form frei programmierbarer Frequenzumrichter oder Mikrocontroller, werden in der Logistik immer häufiger eingesetzt. Aber trotz modularer Mechanik und gesteigerter Rechenleistung anlagennaher Komponenten wird der größte Teil der Entscheidungen immer noch vom Materialflussrechner getroffen. Aber gerade der

Einsatz zentraler, hoch komplexer Steuerungsarchitekturen ist einer der wichtigsten Gründe, warum die Technik kaum noch mit den Anforderungen Schritt halten kann. Denn die anhaltenden Trends in Richtung individualisierter Produkte, sinkender Losgrößen und sehr dynamischer Absatz- und Beschaffungsmärkte fordern der Logistik ein hohes Maß an Flexibilität und Wandelbarkeit ab.

Der Lehrstuhl für Fördertechnik Materialfluss Logistik (fml) der TU München entwickelt im BMBF-Forschungsverbund „Internet der Dinge“ [BMBF] zusammen mit namhaften Partnern aus Industrie und Forschung (Fraunhofer IML, Lanfer, PSI, Schmalz, Siemens, Stöcklin, swisslog, viastore) eine dezentrale Materialflussteuerung, die sich am Vorbild des Internets orientiert. Diese zeichnet sich durch hohe Robustheit und Wandelbarkeit aus, setzt sie doch auf flache Hierarchien aus autonomen, sich selbständig vernetzenden Einheiten. Aus herkömmlichen Baukastensystemen bekannte Module, wie z.B. Weichen, Förderer oder Elektrohängebahnkatzen übernehmen Aufgaben wie Auftragsdisposition, Routing, Reservierung und Bewertung von Wegen, Ressourcenmanagement und Optimierung und erfüllen diese autonom (siehe Abb. 1). Durch den Wegfall des Materialflussrechners sinkt die Systemkomplexität und die Wandelbarkeit und Reaktionsfähigkeit verbessern sich stark [Wil-06].

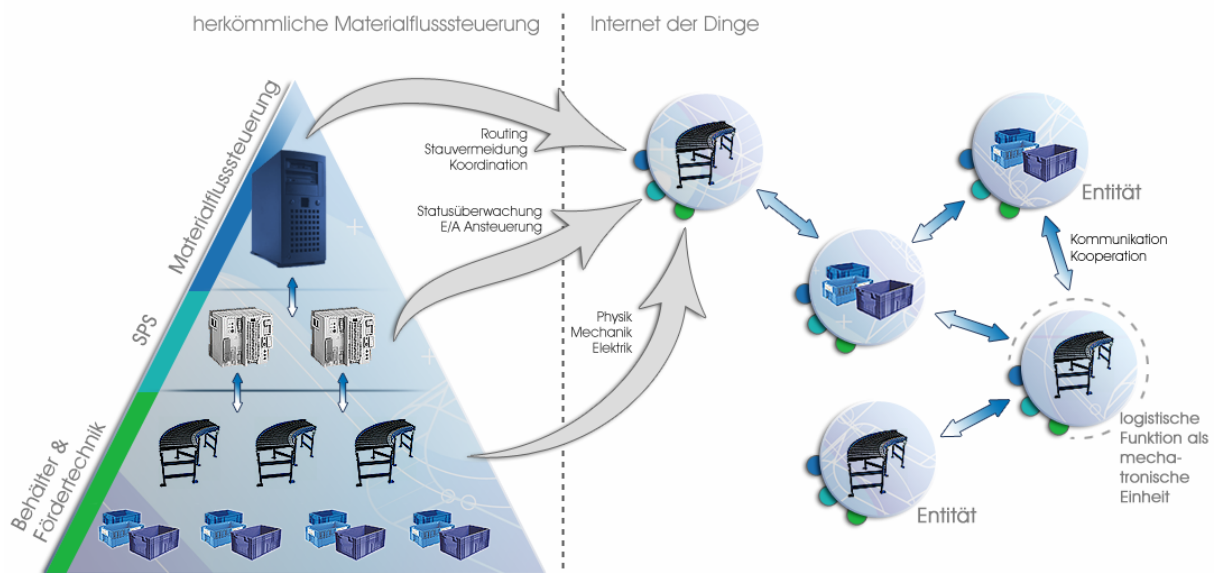


Abbildung 1 Das Internet der Dinge ist ein hierarchieloses Materialflusssystem aus autonomen, intelligenten und kooperierenden Einheiten. Jede Einheit übernimmt Aufgaben aus allen Ebenen heutiger Steuerungshierarchien.

Die technischen Grundlagen, die für die Realisierung eines solchen Systems aus autonomen Einheiten notwendig sind, werden in der Informatik schon seit längerem untersucht und mit Erfolg eingesetzt. Dort verfolgen autonome Softwareprogramme, so genannte Agenten, in ihrem virtuellen Umfeld verschiedene Ziele und kooperieren oder konkurrieren miteinander, ohne zentral gesteuert zu werden. Das wohl bekannteste, wenn auch sehr einfache Beispiel für solche Programme, sind Bietagenten bei Online-Auktionsplattformen – ihre Aufgabe besteht darin, einen ausgewählten Artikel möglichst günstig zu ersteigern, ohne dabei das vom Benutzer festgelegte Höchstgebot zu überschreiten. Agenten können aber auch viel komplexere Aufgaben übernehmen [Rit-03].

Eine der wichtigsten Institutionen, die sich mit der Entwicklung und Spezifizierung solcher Multiagentensysteme befasst, ist die Foundation for Intelligent Physical Agents. Als Mitglied des IEEE erarbeitet diese Institution Vorschläge und Standards zu zahlreichen Themen [FIPA]. Software-Frameworks, wie z.B. das frei verfügbare Java Agent Development Framework [JADE], implementieren diese Standards und liefern die Grundlage für die Entwicklung anwendungsspezifischer Agenten. Die gesamte Basisfunktionalität, z.B. Kommunikationsprotokolle und Managementfunktionen, ist also bereits vorhanden. Bei der Entwicklung eines agentenbasierten Materialflusssystemes kann man auf diese Werkzeuge zurückgreifen und kann seine Aufmerksamkeit komplett der Entwicklung von Algorithmen für die Steuerung des Materialflusses widmen. Dies senkt den Aufwand für Programmierung und Debugging erheblich.

Anders als in der IT gibt es im Internet der Dinge aber nicht nur Softwareprogramme, sondern auch Fördertechnikmodule und Transporteinheiten. Die softwaretechnische Umsetzung muss die zusätzlichen Anforderungen dieser autonomen Einheiten oder „Entitäten“ berücksichtigen.

Steuerungsarchitektur eines Moduls

Im Internet der Dinge können die Aufgaben eines Fördertechnikmoduls in zwei Bereiche gegliedert werden. Zum Einen die Ansteuerung der Mechanik, die vornehmlich auf SPSen umgesetzt wird, und zum Anderen die strategische Entscheidungsfindung und Abstimmung mit anderen Entitäten im Gesamtsystem,

also die Funktionen, die heutzutage hauptsächlich vom Materialflussrechner übernommen werden.

Im anlagennahen Bereich werden heute fast durchgängig die IEC-61131-3 Programmiersprachen eingesetzt. Diese sind eine sehr gute Möglichkeit, um Ein- und Ausgänge zu überwachen und zu logischen Schaltkreisen zusammenzufassen. Die von einer SPS garantierten Taktzeiten machen es möglich, schnell ablaufende Prozesse zuverlässig zu steuern und Sicherheitsbestimmungen einzuhalten. Im Internet der Dinge erledigen Module aber auch viele komplexe Aufgaben wie Wegplanung, Koordination und Kommunikation über Internetprotokolle. Dafür eignen sich objektorientierte Programmiersprachen aus dem PC-Bereich aber viel besser als SPS-Sprachen, zudem in diesem Bereich keine so harten Echtzeitanforderungen gelten.

Die Steuerungslogik eines Moduls wird daher aus zwei Schichten aufgebaut (siehe Abb. 3). Der bspw. in C# oder Java programmierte Softwareagent übernimmt die Kommunikation mit anderen Entitäten und alle dispositiven und strategischen Entscheidungen. Die echtzeitfähige, in herkömmlichen SPS-Sprachen umgesetzte, Maschinensteuerungsebene ist dabei in jedem Modul gekapselt und hat, neben Möglichkeiten zur Diagnose oder dem Debugging von Programmen, keinerlei Schnittstellen nach außen. Dies hat den Vorteil, dass hersteller- und hardware-spezifische Details nicht in den Softwareagenten gelangen, was die Wiederverwendbarkeit der Agentenlogik gewährleistet.

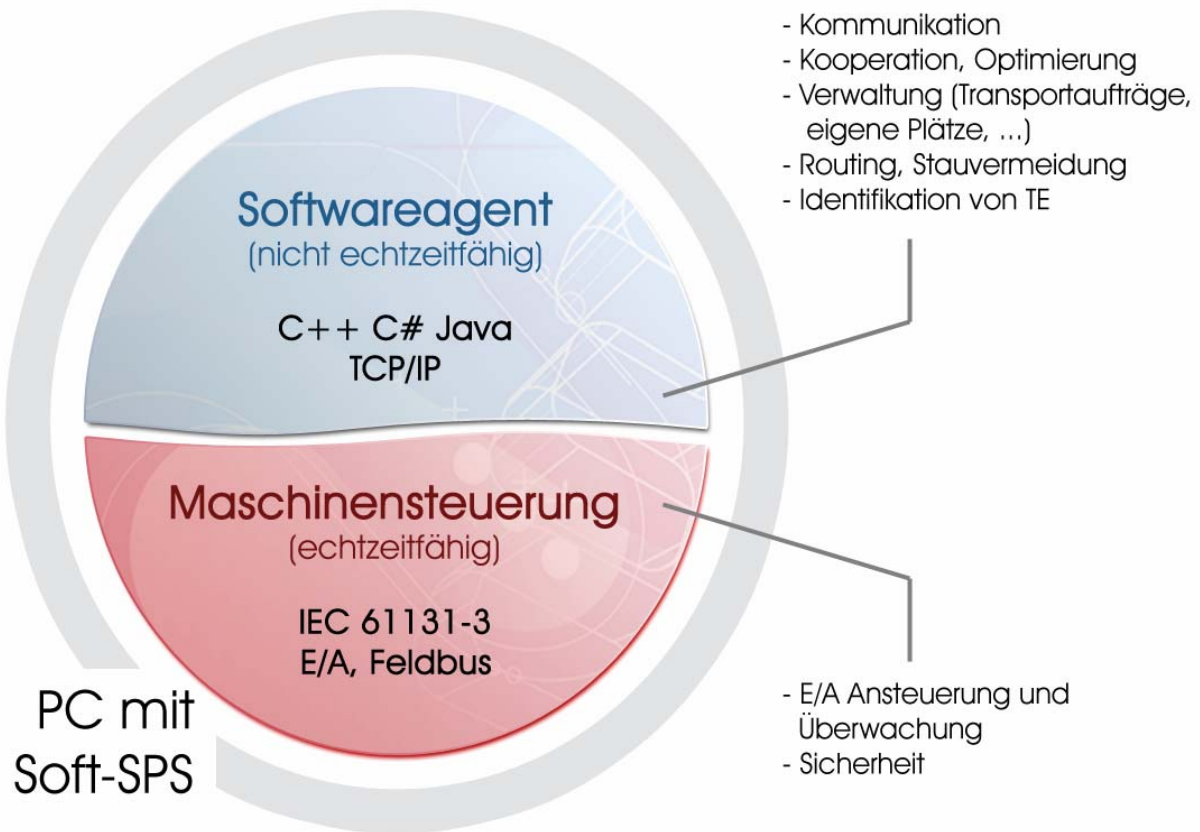


Abbildung 2 Die zwei Schichten der Steuerungslogik eines Moduls

Als Hardwareplattform für eine so gestaltete Steuerungslogik eignen sich Industrie-PCs, die sowohl mit einem gängigen Betriebssystem als auch mit einer Soft-SPS ausgestattet sind. Diese Geräte sind bereits seit Jahren zu Preisen um die 600€ am Markt verfügbar.

Die zur Kommunikation zwischen Softwareagent und Maschinensteuerung notwendigen Protokolle werden ebenfalls in eine eigene Schicht ausgelagert (siehe Abb. 4). Diese kann als externe Bibliothek gestaltet werden und bietet eine einfache Schnittstelle zum Lesen und Schreiben von SPS-Variablen. Der Agent interagiert dann nur mit dieser Middleware-Schicht, und muss nicht an die internen Kommunikationsprotokolle angepasst werden. Der Agent teilt der Middleware mit, welche Protokolle sie zu verwenden hat – die weitere Konfiguration geschieht automatisch.

Der Agent, der im Normalfall den Großteil der Modullogik enthält, kann so hardwareunabhängig ausgeführt und somit auf verschiedenen Geräten eingesetzt werden. Auf diese Weise kann die Ebene der Maschinensteuerung jederzeit ausgetauscht werden, ohne dass eine Anpassung des Agenten notwendig wird.

Ebenso kann die Maschinensteuerungsebene durch einen Emulator ersetzt werden, sodass der Funktionstest eines Moduls durchgeführt werden kann, noch bevor die IEC 61131-3 Programme erstellt oder die Hardware verfügbar ist.

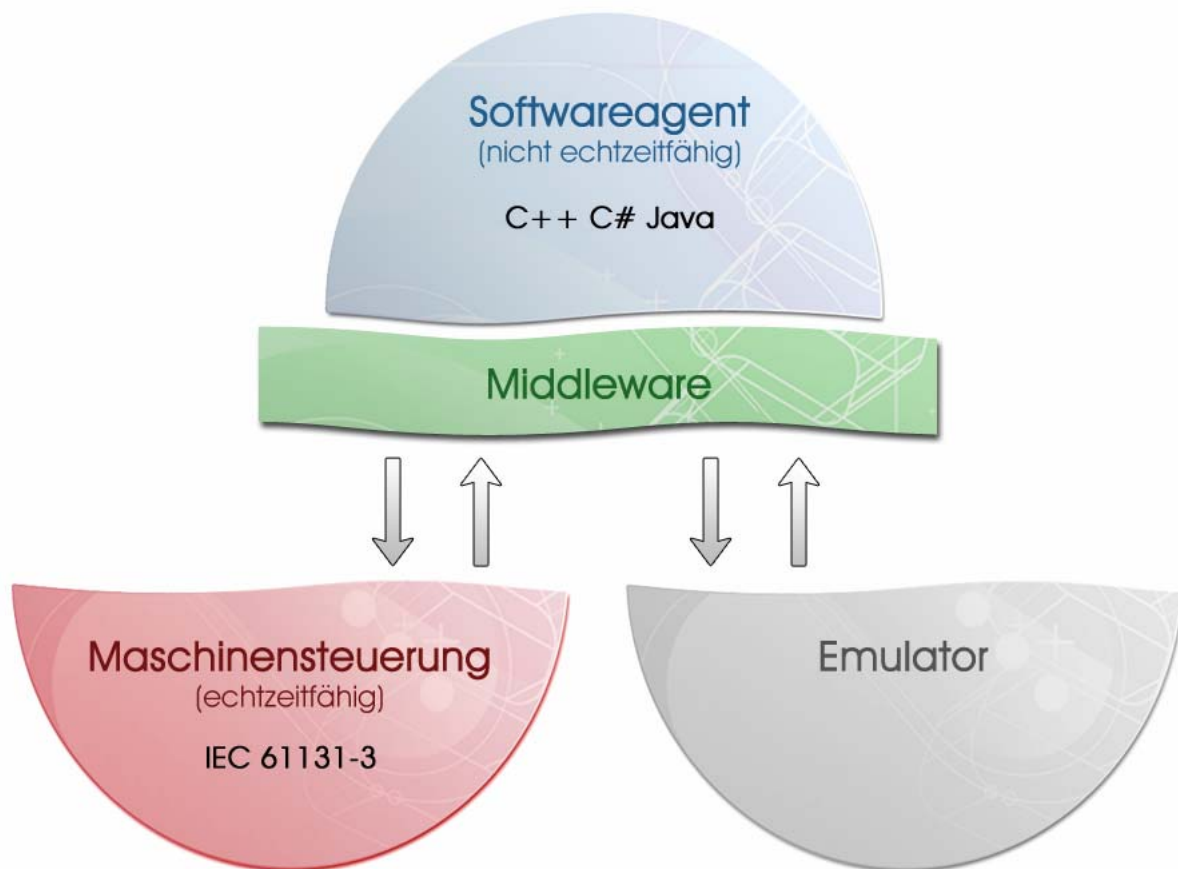


Abbildung 3 Einsatz einer Middleware-Schicht zur Entkopplung von Softwareagent und Maschinensteuerung

Vererbung und Generierung neuer Agenten

Durch den Einsatz höherer Programmiersprachen für die Realisierung eines Großteils der Modulsteuerung können die Vorteile und Potenziale der Objektorientiertheit auch im anlagennahen Bereich genutzt werden.

Als Grundlage für alle im Internet der Dinge benötigten Agenten dient ein „Entitäts“-Agent. Dieser implementiert nur die grundlegendsten Funktionen, die von allen Systemteilnehmern erfüllt werden, wie z.B. Kommunikation oder die Fähigkeit, anhand verschiedener Kriterien nach anderen Entitäten zu suchen. Davon werden Agenten für die drei verschiedenen Entitätstypen – Module, Transporteinheiten und Softwaredienste – abgeleitet, die bereits einen höheren Spezialisierungsgrad aufweisen. Eine Grundfunktion von Diensten könnte eine GUI sein, die es einem

Bediener erlaubt, Daten über das Gesamtsystem einzusehen oder im Extremfall steuernd einzugreifen. Ein Transporteinheits-Agent wird hingegen in der Lage sein, Fahraufträge an Module zu erteilen oder Nachrichten von RFID-Readern zu verarbeiten, um so den Aufenthaltsort eines Behälters in der Anlage zu verfolgen. Der Agent für Module enthält die erwähnte Middleware-Schicht zur Kommunikation mit der Maschinensteuerung, aber noch keine Steuerungslogik für bestimmte fördertechnische Aufgaben, wie z.B. Lastwechsel oder Wegplanung. Diese wird wiederum in einem nächsten Schritt hinzugefügt. Beispielsweise wird ein generischer Agent für Verzweigungs- und Zusammenführungsmodule vom Grundmodul vererbt und implementiert selbst nur noch die zum Schalten von Wegverbindungen notwendigen Funktionen (s. Abbildung 4). Von dem Verzweigungsmodul können Weichen, Drehtische oder Verschiebewägen abgeleitet werden.

Auf diese Weise wird jeder Detaillierungsgrad einer logistischen Funktion schrittweise und auf bereits implementierte Softwarebausteine aufbauend hinzugefügt, was den Programmieraufwand für neue Entitäten stark begrenzt. Sollen grundlegende Mechanismen, beispielsweise die Kommunikation, an neue technologische Möglichkeiten angepasst oder ihre Funktionalität erweitert werden, muss wiederum nur an einziger Stelle umprogrammiert werden. Änderungen an der Entität werden automatisch über das gesamte Softwareprojekt vererbt, ohne dass auch nur eine Codezeile in anderen Agenten verändert werden muss.

Das Internet der Dinge erlaubt also nicht nur die Gestaltung wandelbarer Logistiksysteme, sondern ist auch in sich selbst erweiterbar und zukunftsfähig. Programmierer von Materialflusssteuerungen müssen nicht mehr, aus Angst vor unvorhersehbaren Folgen einer neuen Softwarekomponente oder vor unverträglichem hohem Programmieraufwand auf die neuesten Technologien verzichten.

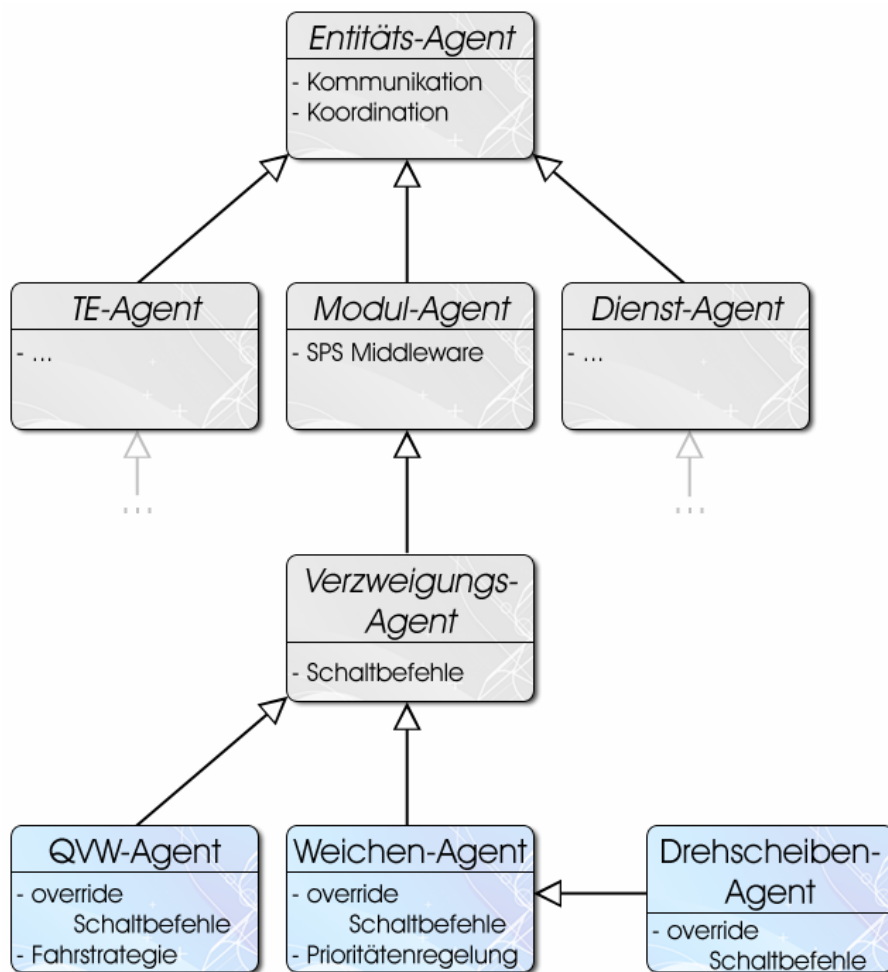


Abbildung 4 Vererbung von Agenten. Abstrakte Klassen (also solche, die nur als Grundlage für andere Klassen dienen) sind grau eingezeichnet

Kommunikation

Wird ein Gespräch zwischen mehreren Menschen betrachtet, kann beobachtet werden, dass dabei prinzipiell zwei verschiedene Kommunikationsmöglichkeiten zum Einsatz kommen. Zum Einen ist das direkte Gespräch zwischen zweien oder wenigen Partnern möglich, zum Anderen kann eine gemeinsamen Plattform, z.B. eine Projektionsleinwand, genutzt werden. In einem Netzwerk autonomer Softwareagenten stehen ebenfalls diese zwei Kommunikationswege zur Verfügung. Dabei haben beide Möglichkeiten charakteristische Vor- und Nachteile: so sind z.B. die Aspekte Systemrobustheit, Kommunikationslast, Verlässlichkeit der Daten oder die gebündelte Verfügbarkeit systemweiter Informationen sehr unterschiedlich ausgeprägt [Sch-03].

Um den hohen Anforderungen an eine Materialflusssteuerung gerecht zu werden, verwendet das Internet der Dinge eine hybride Kommunikationsform: Informationen, die zwischen einer festen, bekannten Gruppe von Entitäten ausgetauscht werden, wie z.B. Schaltaufträge für Weichen oder Koordinationsdaten beim Lastwechsel, werden direkt ausgetauscht. Anders werden Daten behandelt, die für eine größere Anzahl evtl. unbekannter Entitäten relevant sind, bzw. die global zur Verfügung stehen müssen. Dazu zählen beispielsweise die Anlagentopologie, Wegreservierungen oder eine Liste der zu erfüllenden Transportaufträge. All diese Daten werden auf einem Blackboard abgelegt und können von allen anderen Agenten gelesen oder verändert werden. Auf diese Weise lassen sich auch systemübergreifende Funktionen, wie z.B. die Visualisierung oder Erhebung von statistischen Daten, sehr einfach durchführen. Da ein einziges Blackboard im Hinblick auf die Robustheit des Systems als sehr kritisch einzustufen ist, können mehrere, parallel arbeitende Blackboards eingesetzt werden. Synchronisationsmechanismen gewährleisten die Konsistenz der Daten über mehrere solcher Knotenpunkte hinweg.

Doch die beste Kommunikationsinfrastruktur nützt nichts, wenn die Teilnehmer nicht die selbe Sprache sprechen. Ein wichtiger Aspekt ist also die Definition geeigneter Schnittstellen und Datenformate. Auch hier lohnt ein Blick in die IT-Welt – denn dort wird die Extensible Markup Language (XML) eingesetzt, um Inhalte in einem standardisierten, erweiterbaren und sowohl von Menschen als auch Maschinen lesbaren Format darzustellen. Eine Vielzahl von Tools und Erweiterungen haben XML zu einem universell einsetzbaren Werkzeug für Kommunikation, Datenspeicherung und –verarbeitung gemacht. Auch für das Internet der Dinge eignet sich XML hervorragend – denn es bietet eine flexible, standardisierte und etablierte Möglichkeit, die Inhalte beliebiger Nachrichten darzustellen und zu strukturieren. Technologien wie Ethernet oder WLAN bieten zu günstigen Preisen ausreichend Bandbreite und sind auch für den industriellen Einsatz mittlerweile sicher genug.

Fazit

Das Internet der Dinge stellt einen neuen Ansatz in der Materialflusssteuerung dar. Aufbauend auf den aus der Informatik bekannten Multiagentensystemen und

verfügbaren Softwareframeworks erlaubt ein Zwei-Ebenen-Konzept den Fördertechnikmodulen Echtzeitvorgänge zu überwachen und gleichzeitig hochkomplexe Entscheidungen zu treffen. Der Einsatz objektorientierter Sprachen ermöglicht es dabei, Steuerungslogik mit minimalem Aufwand für verschiedene Module wieder zu verwenden, was den Implementierungsaufwand senkt und die Robustheit der Software erhöht. Das hybride Datenaustauschprinzip, das sowohl direkte Kommunikation als auch die Nutzung eines gemeinsamen Datenpools beinhaltet, ermöglicht sowohl die effiziente Kooperation und Koordination innerhalb des Netzwerkes als auch übergreifende Funktionen wie Visualisierung und Optimierung.

Diese Kombination zahlreicher und bewährter Technologien und Standards macht das Internet der Dinge zu einer leistungsfähigen und robusten Materialflusssteuerung. Die anforderungsgerechte Auswahl von Rechen- und Kommunikationssystemen für die verschiedenen Bereiche der Anlagensteuerung sowie die flexible und zukunftsfähige Softwarearchitektur gewährleisten sowohl die Effizienz des Systems als auch seine Offenheit gegenüber zukünftigen, leistungsfähigeren Technologien. Dadurch ist das Internet der Dinge nicht nur der Schlüssel zu mehr Wandelbarkeit in der Logistik, sondern stellt eine in sich selbst wandelbare Architektur dar. Lesen Sie im nächsten Artikel dieser Reihe über die wirtschaftlichen Potenziale dieser innovativen Technologie.

[FIPA-05] FIPA Standards Committee (FIPA SC), Policies and Procedures; Institute of Electrical and Electronics Engineers (IEEE), 2005

[JADE-07] Fabio Luigi Bellifemine, Giovanni Caire, Dominic Greenwood: Developing Multi-Agent Systems with JADE; Wiley, ISBN: 978-0-470-05747-6, 2007

[BMBF-06] Das Forschungs- und Entwicklungsprojekt "Internet der Dinge" 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.

[Wil-06] Wilke, M.: Wandelbare automatisierte Materialflusssysteme für dynamische Produktionsstrukturen, Diss., TU München; Herbert Utz Verlag München, ISBN 3-8316-0591-2, 2006

- [Sch-03] Schahram, D.; Gall, H.; Hauswirth, M.: Software-Architekturen für Verteilte Systeme, Springer Verlag, Juli 2003
- [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