

Dezentrale Steuerung für Materialflusssysteme am Beispiel von Stückgutförder- und sortieranlagen

PROF. DR. M. TEN HOMPEL,
FRAUNHOFER INSTITUT FÜR MATERIAFLUSS UND LOGISTIK (IML), DORTMUND

DIPL.-ING. S. LIBERT; DIPL.-ING. (FH) U. SONDHOF
LEHRSTUHL FÖRDER- UND LAGERWESEN, UNIVERSITÄT DORTMUND

Der Lehrstuhl Förder- und Lagerwesen der Universität Dortmund hat zusammen mit einem Industriepartner ein neuartiges, dezentrales Steuerungssystem entwickelt. In diesem modularen, hochflexiblen System wird der gesamte Materialfluss durch dezentral organisierte, autonome Einheiten in Echtzeit gesteuert. Das System wurde erfolgreich in eine am Lehrstuhl betriebene Stückgutförder- und Sortieranlage integriert, welche als Prototyp dient und für weitere Forschungen im Bereich dezentrale Automatisierung zur Verfügung stehen wird. Dieses Dokument beschreibt die Zielsetzung und Implementierung des dezentralen Steuerungssystems.

1. Stand der Technik

Die sich verbessernde Leistung von Rechnersystemen und das große Spektrum an Applikationen ermöglichen neue, effiziente Lösungen für viele „konservative“ Bereiche der Industrie. Sowohl Benutzer als auch Entwickler von Steuerungssystemen in Produktion und Logistik können von solchen Lösungen profitieren. Eine der in diesem Zusammenhang zurzeit immer öfter diskutierten Aufgaben ist die Entwicklung neuartiger, intelligenter Steuerungssysteme, die eine Selbststeuerung logistischer Prozessen in der Zukunft ermöglichen [Freitag04]. Diese Systeme werden auf verteilten bzw. dezentralen Steuerungsarchitekturen basieren [tenHompel04]. Der große Vorteil dezentraler Architekturen ist die Möglichkeit, hochflexible Systeme zu erschaffen, welche sich auf einfache Art und Weise an neue Betriebsituationen anpassen können [Günthner02]. Dadurch werden der Systemauf- und -umbau beschleunigt sowie Kosten für eine aufwändige Implementierung reduziert. Der Einsatz dezentraler, autonomer Steuerungen, basierend auf kostengünstigen und intelligenten Rechnersystemen, spielt somit eine ähnliche Rolle wie die Einführung von Feldbussystemen vor einigen Jahren.

Die klassischen Steuerungssysteme bestehen in der Regel aus einer zentralen Speicherprogrammierbaren Steuerungen (SPS), installiert in einem großen Schaltschrank mit mehreren Sicherungen und Relais (vgl. [tenHompel03], [Wellenreuther98]). Sensoren und Aktoren werden mit der SPS entweder Punkt-zu-Punkt oder per Feldbus verknüpft. Dabei wird durch die SPS für jeden Sensor oder Aktor ein eigener Eingangs-/Ausgangskanal (E/A-Kanal) bereitgestellt. Sämtliche Kanäle werden von einer Rechneinheit der SPS bearbeitet, die schnell genug sein muss, um die Echtzeitanforderung (Antwortzeit ca. 20 ms) für jeden der zahlreichen E/A-Anschlüsse zu erfüllen.

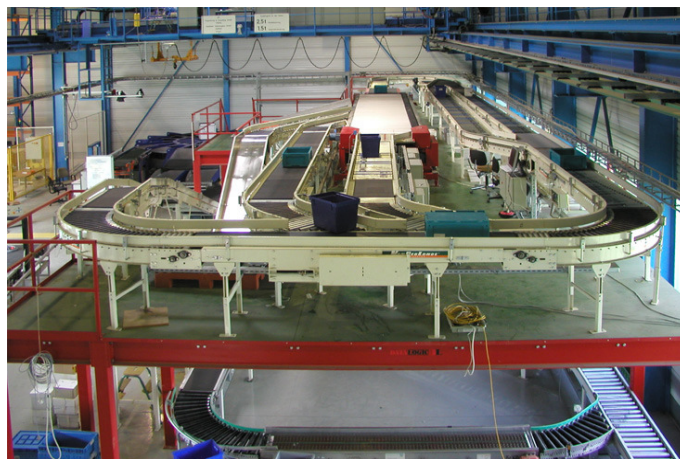


Abb. 1 Förder- und Sortieranlage

Die Programmierung solcher SPS erfolgt zumeist nach dem IEC1131-Standard in einer Assembler-ähnlichen oder auch in einer höheren Programmiersprache. Das Programm selbst besteht i.A. aus einer Hauptschleife, die sequentiell alle E/A-Kanäle bearbeitet. Die SPS sind dabei für hohe Geschwindigkeit optimiert, so dass Programmdurchläufe von wenigen Millisekunden erreicht werden. Eine solche auf SPS basierende, zentralisierte Topologie ist allerdings relativ unflexibel. Die Softwareanpassung ist aufgrund der hohen Komplexität zentraler Steuerungen nur unter hohem Arbeitsaufwand durchführbar. Änderungen erfordern normalerweise die gesamte Überarbeitung des Steuerungsprogramms. Dies hat zur Folge, dass solche Systeme über lange Zeit unverändert in Betrieb bleiben müssen.

Am Lehrstuhl Förder- und Lagerwesen existiert eine Förderanlage (siehe Abbildung 1), die in der Vergangenheit von einer zentralen SPS gesteuert wurde. Es wurde ein Pilotprojekt durchgeführt, mit dem Ziel, eine neuartige, dezentrale Steuerung zu entwickeln, die in die vorhandene Förderanlage integriert werden sollte

2. Genereller Ansatz und Zielsetzung

Die konsequente Modularisierung des Systems war zentraler Punkt des Projektes. Um dies zu erreichen, müssen die mechanischen Segmente mit Steuerungssoftware kombiniert werden, so dass einheitliche und austauschbare, funktionale Module entstehen. Dieses Modularitätsprinzip führt zu einem Baukastenprinzip für die Hersteller von Materialflusssystemen und verringert die Komplexität von Steuerungssystemen. Weiterhin kann ein Teil der Aufgaben für die Inbetriebnahme einer Anlage schon während der Planungsphase erfüllt werden.

Eine andere Aufgabe, die von dem neuen Steuerungssystem erbracht werden sollte, ist die Verteilung der Rechenleistung auf viele dezentrale Steuerungsknoten. Dies trägt dazu bei, dass die Skalierbarkeit und die Gesamtleistung der Steuerung erhöht werden.

Ein weiteres Ziel des Vorhabens war die Erschaffung eines zukunftsweisenden, offenen und intelligenten Steuerungskonzeptes, das Standards des Internet wie z. B. TCP/IP, HTTP oder XML für die Kommunikation und Datenübertragung verwendet.

Für die Ermöglichung eines solchen Konzeptes mussten alle Teilnehmer eines Materialflusses in das Kommunikationsnetzwerk integriert werden: der Lieferant am Anfang, der Kunde am Ende und das automatisierte Materialflusssystem in der Mitte der Kommunikationskette.

Der erste Schritt zur Realisierung des Modularitätsprinzips war die Definition funktionaler Module. In Abbildung 2 ist der technologische Aufbau der unteren Ebene (Ebene 1) der Testanlagen vereinfacht dargestellt. Die Förderanlage besteht aus verschiedenen automatisierten mechanischen Segmenten (z. B. Gurt- oder Rollenförderer, Pufferstrecken, Weichen, Zusammenführungen usw.). Jedes dieser Fördersegmente benötigt eine entsprechend ausgelegte Softwaresteuerung. Als Hardware-Plattform (Knotenrechner) sollten kompakte Industrie-PC (IPC) verwendet werden, auf denen zum Teil auch mehrere autonom arbeitende Softwaresteuerungen gleichzeitig laufen können. Die Zuweisung der fördertechnischen Segmente mit entsprechenden Steuerungsprogrammen zu den Knotenrechnern wird für die Ebene 1 in Abbildung 3 dargestellt.

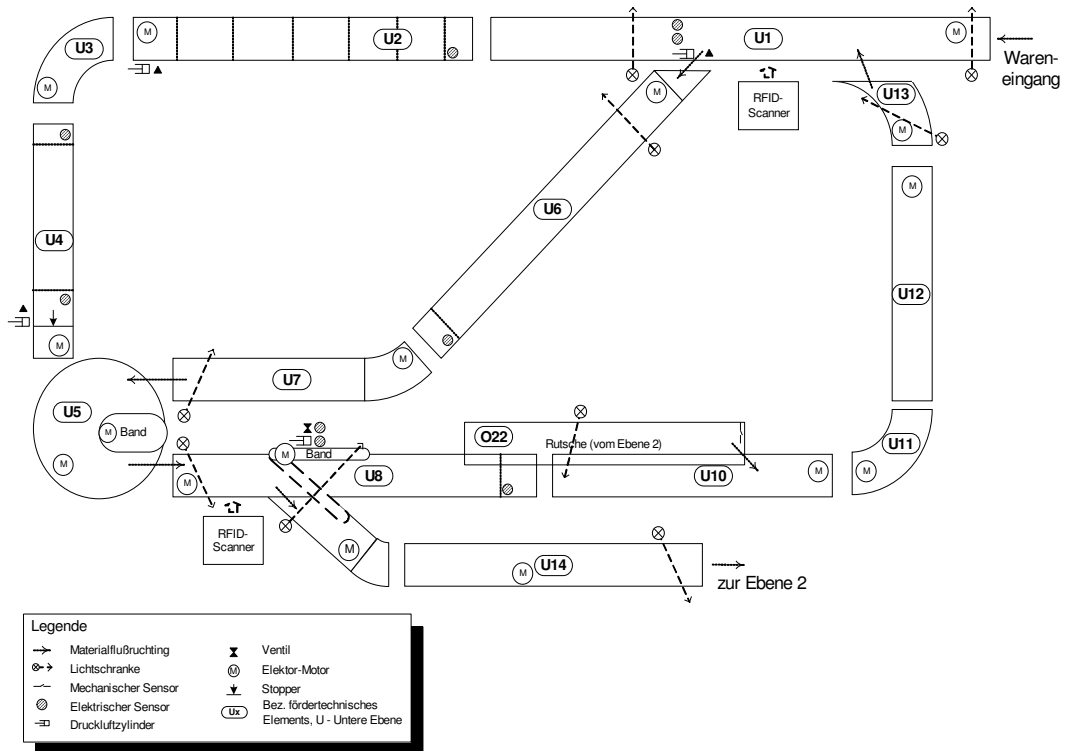


Abb. 2 Layout der Förderanlage (Ebene 1)

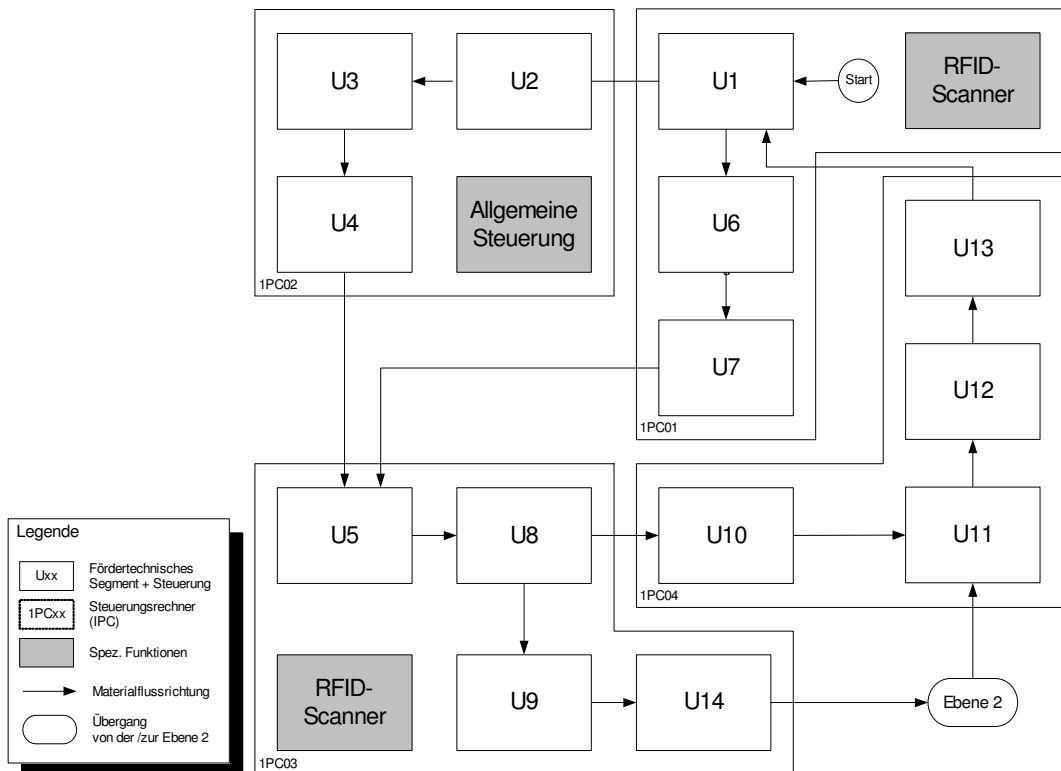


Abb. 3 Zuweisung der Förderelemente zu den Knotenrechnern (Ebene 1)

3. Netzwerk- und Plattformspezifikation

Die Netzwerkstruktur besteht aus einer Vernetzung von sieben Knotenrechnern und zwei Arbeitsplatz-PC per Ethernet (Abbildung 4). Der Einsatz von Ethernet ist im Vergleich zu Feldbus-systemen kostengünstiger und bietet gleichzeitig eine hohe Skalierbarkeit. Dabei ermöglicht der

Einsatz des TCP/IP-Protokolls eine transparente Kommunikation sowie standardisierte, einheitliche Kommunikationsschnittstellen.

Die Arbeitsplatz-PC und die Knotenrechner agieren im gleichen Subnetz. Hier besteht noch ein Risiko bezüglich der Verfügbarkeit und Stabilität der Kommunikationszugriffe. Diese Konfiguration ist für einen Systemprototypen mit relativ niedriger Systemkomplexität und Kommunikationssicherheitsanforderungen ausreichend. In Industriebereichen ist es allerdings sinnvoll, das Steuerungsnetz von Arbeitsplatz-PC durch einen Gateway-Router zu trennen, um eine sichere Stabilität und eine hohe Datenübertragungsleistung zu gewährleisten.

Es ist dabei zu beachten, dass die Anforderungen zur Datenübertragung in solchen Systemen im Allgemeinen sehr hoch sind. In einem dezentral gesteuerten System besteht kein Bedarf die Prozess- und Steuerungsdaten mit einem übergeordneten Materialflusssteuerungssystem zu tauschen. Gerade diese Unabhängigkeit erfordert jedoch Absprachen unter den autonomen Steuerungsprogrammen und resultiert in einer erhöhten Kommunikation im Vergleich zu zentralisierten Systemen: Sobald ein physikalischer Materialfluss entsteht, werden auch sein digitales Äquivalent und die Zustandsdaten von einem Teilnehmer zum anderen durch das Netzwerk transferiert.

Die zwei in das System integrierten Standardrechner erfüllen folgende Funktionen (s. Abbildung 4):

1. Client für die Konfiguration und Visualisierung der Anlage
2. Datenschnittstelle für einen Lieferanten/Kunden (EDI¹-Client)

Ein dritter Rechner repräsentiert den EDV-Arbeitsplatz eines Lieferanten und wird für die Abwicklung der Datenkommunikation via Internet mit dem Identifikationspunkt der Förderanlage verwendet.

Die Hardwareplattform für die Steuerungssoftware muss ein kompaktes System sein, das multitaskingfähig ist und eine Echtzeitunterstützung bietet. Für die momentane Implementierung wurden Industrie-PC (IPC) und ein Betriebssystem mit Echtzeitunterstützung gewählt. Die verwendeten IPC sind kompakt und haben eine geringe, aber erweiterbare Anzahl an E/A-Anschlüssen. Diese auf PC-Architektur basierenden Rechner sind in einem speziellen Gehäuse untergebracht, das eine schnelle Hutschienenmontage ermöglicht. Es sind weder Bildschirm, noch Maus oder Tastatur vorhanden. Die gesamte Interaktion, Datenkommunikation und Konfiguration erfolgt per Netzwerk (Ethernet).

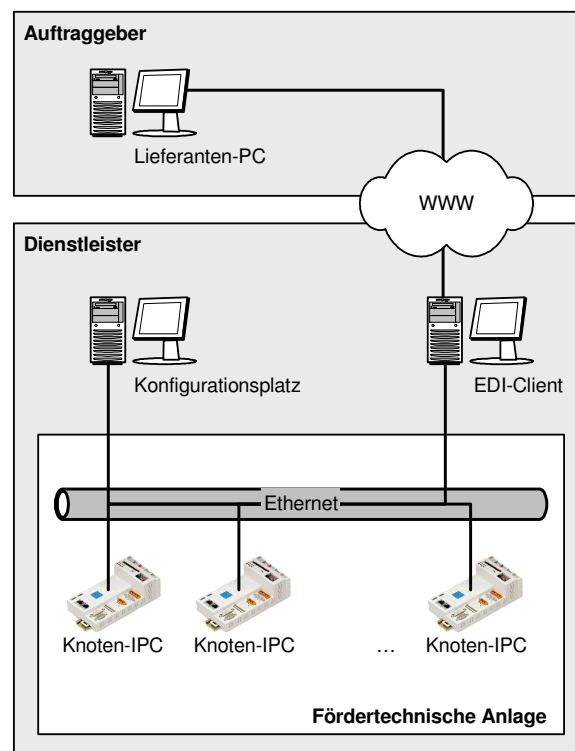


Abb. 4 Netzwerkstruktur

Das eingesetzte Betriebssystem ist Linux mit RTAI (Real Time Application Interface) - eine Echtzeit-Erweiterung für den Linux-Kernel (s. z. B. [Walter01], [Schwebel01]). Dieses System

¹ EDI – Electronic Data Interchange

wurde entwickelt in Übereinstimmung mit den Richtlinien für freie Software und entspricht den Eigenschaften eines industriellen RTOS (Real Time Operating System).

4. Konzept der Steuerungssoftware

Eine hohe Modularität war die Hauptanforderung an das Softwaredesign der dezentralen Steuerung. Dies wird erreicht, wenn jedes mechanische Segment (z. B. Förderer oder Sorter) durch ein eigenes Steuerungsprogramm gesteuert wird. Eine solche Software sollte die Funktionalität der darunter liegenden mechanischen Teile repräsentieren und idealerweise auf einem separaten Knotenrechner laufen. Die Anzahl solcher dezentralen, kooperierenden Softwareeinheiten ist dann gleich der Anzahl der mechanischen Fördersegmente. Dabei bilden der mechanische Teil (Förderelement) und der Automatisierungsteil (Sensoren und Aktoren) zusammen mit dem Steuerungsteil (Rechner und Steuerungssoftware) ein einheitliches, funktionales Modul mit einheitlichen mechanischen sowie kommunikativen Schnittstellen. Sind diese Schnittstellen systemweit uniform, so ist ein Austausch der funktionalen Module oder die Erweiterung des Systems relativ einfach.

Im Zuge der Entwicklung eines dezentralen Steuerungssystems gibt es folgenden Widerspruch zu klären: zum einen sollten die Softwareeinheiten uniform und unabhängig von der Mechanik und der Automatisierungstechnik sein. Zum anderen bestehen die realen Anlagen oft aus einer großen Anzahl sehr unterschiedlicher Fördererelemente. Um diesem Widerspruch zu begegnen, besteht die entwickelte Software aus zwei wichtigen Teilen: Jedes Steuerungsprogramm (eins für jedes Fördersegment) beinhaltet sowohl einen hardwareunabhängigen Teil (logische Komponente) als auch einen E/A-Treiber (s. Abbildung 5). Der Treiber hängt somit von der darunter liegenden Förder-technik ab. Er verarbeitet die Sensorsignale und sendet Befehle an die Aktoren. Da dies durch einen mit RTAI erweiterten Linux-Kernel unterstützt wird, kann die Echtzeitanforderung erfüllt werden.

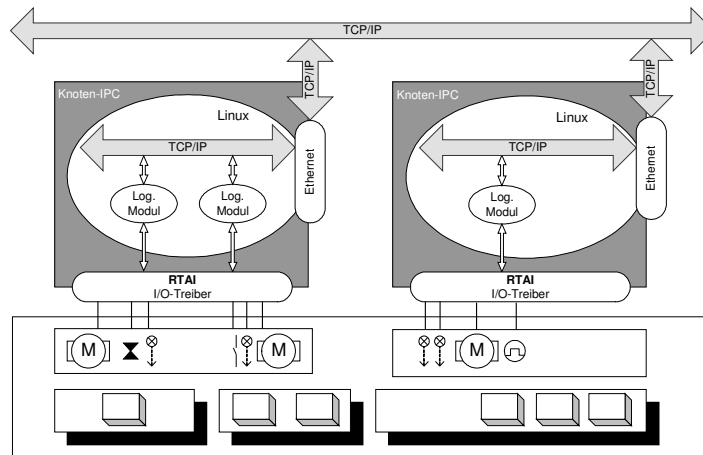


Abb. 5 Übersicht der Systemarchitektur

Im Gegensatz zu den E/A-Treibern ist die Architektur aller hardwareunabhängigen logischen Komponenten („log. Modul“ in Abbildung 5) identisch. Die Konfiguration der verschiedenen Parameter erfolgt zur Laufzeit und wird in XML-Dateien abgelegt. Diese speichern u.a. Informationen über die Adressierung der E/A-Anschlüsse der IPC, die Netzwerk-IP-Adressierung der Knotenrechner und die Beschreibung der Anlagentopologie.

Abbildung 6 zeigt ein Beispiel einer E/A-Konfiguration auf Basis von XML. Das XML-Format ist einfach zu lesen und zu verarbeiten, sowohl von Menschen als auch von Maschinen, dazu ist es noch sehr flexibel und umfangreich. Änderungen an der Systemkonfiguration, die nur von einem Entwickler oder Systemadministrator durchgeführt werden dürfen, können zur Laufzeit vorgenommen werden. Die Übernahme der Konfigurationsdaten in die Steuerung und ihre Verarbeitung, basierend auf SAX-Modell [Shepard02], wird ohne Neukompilieren der Software automatisch durchgeführt. Somit ist ein hohes Konfigurationsmaß gegeben.



```

<IPC Name="UC1" ID="1">
  <Module Name="U1">
    <DigIO Type="INPUT"
Name="BU1.1" Size="BIT"
Addr="0100">
      <Description>Position at
start U1</Description>
    </DigIO>
    ...
    <DigIO Type="OUTPUT"
Name="MU1" Size="BIT" Addr="1000">
      <Description>MU1
on</Description>
    </DigIO>
    ...
  </Module>
  ...
</IPC>

```

Abb. 6 Beispiel einer Konfiguration mit XML

Idealerweise wird jedes Steuerungsprogramm auf einem eigenen IPC gestartet. Aus Kostengründen können aber auch mehrere Programme autonom zueinander auf einem einzigen IPC aktiv sein. Grundsätzlich sind somit viele verschiedene Systemorganisationen möglich - von einer Architektur mit nur einem zentralen Rechner (alle Softwaresteuerungen auf einem IPC) bis hin zu einem komplett dezentralen System (ein IPC für jede Softwaresteuerung).

5. Datenaustausch und Kommunikation

Die Softwaresteuerungen arbeiten autonom und unabhängig von jeglichen zentralen Steuerungssystemen. Bei solchen Bedingungen ist eine schnelle, stabile und transparente Kommunikation ein sehr wichtiger Faktor. Damit ein fortlaufender Materialfluss garantiert werden kann, müssen folgende Anforderungen von der Kommunikation erfüllt werden:

- ein sicherer Informationsaustausch bezüglich des Zustandes der darunter liegenden Fördertechnik zu jeder Zeit muss gegeben sein
- ein sicherer Informationsaustausch bezüglich der Stückgüter und der Aufträge muss garantiert werden
- der Zustand der Steuerung muss abfragbar sein, um Fehlermeldungen zu generieren und eine Visualisierung zu ermöglichen.

Die Visualisierung des Anlagenzustandes erfolgt über den Konfigurations-PC, auf den ein Java-Client installiert ist. Der Client kann die Zustände des Steuerungsprogramms über das HTTP-Protokoll abfragen und diese in einem Web-Browser darstellen.

Der Datenaustausch zwischen den Steuerungsprogrammen erfolgt über TCP/IP-Sockets. Die Art der Kommunikation ist unabhängig davon, ob diese Programme auf demselben oder auf unterschiedlichen Knotenrechnern arbeiten. In den Telegrammen, die von den Softwaresteuerungen ausgetauscht werden, befinden sich zwei Arten von Daten - Statusanfragen sowie Stückgutinformationen.

Die Stückgutinformationen beinhalten die Stückgutidentifikationsnummer und die Auftragsinformation. Diese Telegramme werden immer zu dem Zeitpunkt an die „nachfolgende“ Steuerung gesendet, wenn das reale Stückgut an das entsprechende Fördertechniksegment übergeben wird. Dadurch wird der Umlauf der Stückgüter virtuell repräsentiert und der reale Materialfluss wird digital reproduziert.

Ein wichtiger Bestandteil des hier vorgestellten Steuerungskonzeptes ist der Einsatz der radiofrequenten Identifikations-Technologie (RFID). Sie hilft dabei, einen kontrollierbaren und sinnvollen Materialfluss zu garantieren. Sie wird sowohl für die Identifizierung der Stückgüter als auch für das dynamische Weiterleiten (Routing) verwendet. Die Auftragsinformationen eines Stückgutes beinhalten jeweils u.a. eine Liste von Zielpunkten auf der Förderanlage, die von dem Paket „besucht“ werden müssen (bis zum Warenausgang). Diese Informationen, zusammen mit der ent-

sprechenden Stückgutidentifikationsnummer werden in den RFID-Tags auf jedem Stückgut gespeichert und können von vier RFID-Scannern ausgelesen und neu beschrieben werden.

6. Weiterleitung von Paketen

Es gibt zwei Bedingungen, die erfüllt sein müssen, damit Stückgüter korrekt zum Zielort weitergeleitet werden können. Zum einen müssen die Auftragsinformationen des entsprechenden Stückgutes verfügbar sein. Zum anderen müssen die Steuerungsprogramme eine exakte Kenntnis der Topologie der Förderanlage haben.

Die Topologie des existierenden Materialflusssystemes ist im Vorfeld bekannt. Diese wird mittels XML formal beschrieben, am Konfigurations-PC abgelegt und ist weiterhin für die Softwaresteuerungen via Netzwerk verfügbar. Die formale Definition der Topologie durch XML hat einige Besonderheiten (s. Abbildung 7). Sie beschreibt Wege in der Anlage, dem die Pakete folgen können. Jeder Weg besteht dabei aus einem oder mehreren Strecken, die jeweils den einander folgenden Fördersegmenten der Anlage entsprechen. Außerdem hat jeder Weg einen Start- und einen Zielpunkt. Jedoch werden in dieser Datenstruktur nicht alle möglichen Wege abgelegt, sondern nur die für die Materialflussteuerung relevanten. Diese sind zum Beispiel die alternativen Pfade von einem Verzweigungspunkt (Sorter, Weiche) zu einem bestimmten Ort (Förderstrecke oder Warenausgabe) der Anlage.

In manchen Situationen macht es Sinn, spezielle Zielpunkte zu definieren. Führen beispielsweise mehrere Wege zu einem bestimmten Punkt der Anlage, so kann ein „Standard“-Zielpunkt definiert und benutzt werden („DEF“ für „default“ in Abbildung 7). Dies bedeutet, dass jeder alternative Weg erlaubt ist, um ein Paket zu seinem Zielort zu transportieren.

Da kein zentraler Materialflussrechner vorhanden ist, müssen die dezentralen Steuerungen die Entscheidungen über die Weiterleitung der Pakete selbstständig treffen. Dafür müssen die Steuerungsprogramme folgende Funktionen vorweisen:

- verwalten eigene Routentabelle,
- leiten die Pakete entlang des besten Pfades weiter, sofern alternative Pfade vorhanden sind,
- lesen beim Systemstart die aktuelle Beschreibung der Anlagentopologie aus einer Konfigurationsdatei.

Sollten mehrere mögliche Wege zu einem Zielort führen, ist es nicht mehr ausreichend, nur den Zielort zu kennen. In diesem Fall wird die Entscheidung durch eine Prioritätenvergabe anhand der Routentabelle getroffen. Diese Prioritäten können dabei sowohl statisch als auch dynamisch vergeben werden.

Im Fall einer dynamischen Prioritätenvergabe ändert sich die Belegung der Prioritäten innerhalb der Routentabelle je nach Auslastung der Förderstrecken. Es sind somit immer die aktuellen Informationen bzgl. Auslastung für alle Teilstrecken erforderlich, was die Komplexität der Kommunikation wesentlich erhöhen kann. Die statischen Prioritäten müssen hingegen in die Routentabelle fest eingetragen werden. Eine höhere Priorität erhalten dann die Förderstrecken, die über höhere Kapazität bzw. Durchsatzvermögen verfügen.

Der Einsatz rekonfigurierbarer Routenprioritäten hat einige Vorteile. So können zum Beispiel einzelne Fördersegmente oder komplette Anlagenbereiche aus dem gesamten Materialfluss ausgeschlossen werden. Setzen des Prioritätswertes auf 0 schließt beispielsweise eine fehlerhafte Förderstrecke aus, die in der Zwischenzeit repariert oder ersetzt werden kann.

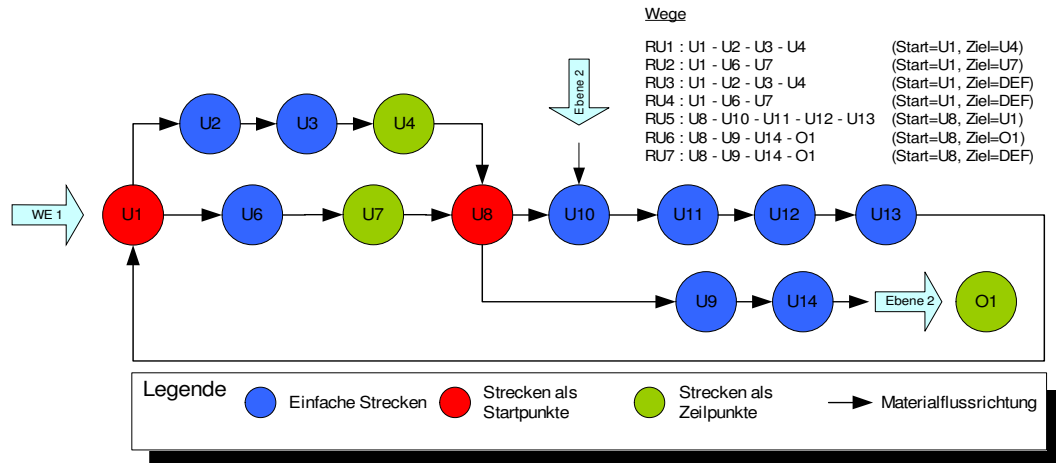


Abb. 7 Beispielbeschreibung der Anlagentopologie (Ebene 1)

7. Fazit

Der Versuch, klassische, zentrale Steuerungen durch intelligente, autonome Einheiten zu ersetzen, ist nur ein weiterer Schritt hin zu modularen und kosteneffizienten Materialflusssystemen. Das dezentrale Steuerungssystem, das am Lehrstuhl Förder- und Lagerwesen entwickelt wurde, demonstriert einen solchen Versuch.

Die Vorteile eines dezentralen Steuerungssystems sind klar erkennbar: Der Einsatz einheitlicher und modularer Software bietet nicht nur eine Erhöhung der Skalierbarkeit, es senkt gleichzeitig die Entwicklungs- und Wartungskosten. Dank einer einheitlichen Busstruktur werden die Kosten und der Installationsaufwand weiter verringert. Zusätzlich wird die Verfügbarkeit des gesamten Steuerungssystems erhöht, indem Konfigurationsanpassungen zur Laufzeit durchgeführt werden können. Diese Eigenschaften führen zu einer kosteneffizienten und flexiblen Systemlösung.

Die Nutzung moderner IT-Standards wie das XML-Format für die Systemkonfiguration und Datenübertragung, das Internet-Protokoll (TCP/IP) für die Datenkommunikation und die RFID-Technologie für die Stückgutidentifizierung machen das vorgestellte Steuerungssystem zu einer guten Basis für die Forschung im Bereich dezentraler Automatisierung und Steuerung. Die Entwicklung dezentraler selbstgesteuerter Systeme geht weiter. Am Ende dieser Entwicklung werden komplexe Materialflusssysteme aus kleinen intelligenten Einzelmodulen zusammengesteckt werden. Eine Programmierung oder Konfigurierung wird nicht mehr nötig sein. Die Softwareagenten übernehmen diese Aufgaben und sind in der Lage, das intelligente Fördergut zum gewünschten Ziel zu bringen. Das *Internet der Dinge* wird Gestalt annehmen. Denn genauso wie die Daten ihren Weg über die Datenleitungen dieser Welt finden, werden es in Zukunft die Pakete tun.

Literatur

- [Freitag04] Freitag, M.; Herzog, O.; Scholz-Reiter, B.: Selbststeuerung logistischer Prozesse – Ein Paradigmenwechsel und seine Grenzen. In: Industrie Management 20 (2004) 1, S. 23-27.
- [Günthner02] Günthner, W.A.; Wilke M.: Anforderungen an automatisierte Materialflusssysteme für wandelbare Logistikstrukturen. In: Tagungsband Wissenschaftssymposium Logistik der BVL 2002, Huss-Verlag GmbH, München, 2002, S. 335-345.
- [Schwebel01] Schwebel, R.: Embedded Linux. mitp-Verlag, 2001.
- [Shepard02] Shepard, D.: XML. Markt+Technik Verlag, 2002, S. 301-326.
- [tenHompel03] ten Hompel, M.; Schmidt, T.: Warehouse Management. Springer, 2003, S. 161-165.
- [tenHompel04] ten Hompel, M.; Stuer, P.: Echtzeitfähigkeit gefordert. In: Fördertechnik 7-8/2004, S. 52-54.
- [Walter01] Walter, K.-D.: Messen, Steuern, Regeln mit Linux. Franzis Verlag, 2001.
- [Wellenreuther98] Wellenreuther, G.; Zastrow, D.: Steuerungstechnik mit SPS. Vieweg Verlagsgesellschaft, 1998.