Simulation in Produktion und Logistik
Entscheidungsunterstützung von der Planung bis zur Steuerung
Wilhelm Dangelmaier, Christoph Laroque & Alexander Klaas (Hrsg.)
Paderborn, HNI-Verlagsschriftenreihe 2013

Ein Ansatz zur Verifikation von Materialflusssystemen durch Model Checking

An Approach to Verification of Material Handling Systems using Model Checking

Karsten Turek, TU Dresden, Dresden (Germany), karsten.turek@tu-dresden.de

Thomas Klotz, Fraunhofer-Institut für Integrierte Schaltungen, Institutsteil Entwurfsautomatisierung, Dresden (Germany), thomas.klotz@eas.iis.fraunhofer.de

Thorsten Schmidt, TU Dresden, Dresden (Germany), thorsten.schmidt@tu-dresden.de

Bernd Straube, Fraunhofer-Institut für Integrierte Schaltungen, Institutsteil Entwurfsautomatisierung, Dresden (Germany), bernd.straube@eas.iis.fraunhofer.de

Abstract: The design of properly working material handling systems (MHS) is a difficult process as these systems consist of a vast number of elements with dedicated controls. While these systems are usually validated using simulation, formal verification using model checking provides alternative means for in-depth analysis of the system behaviour. The approach is based on a modelling methodology considering material handling elements and their controls. The paper discusses the enhancement of discrete-event simulation with this approach, resulting in advanced means to analyse material handling controls, and gives an example.

1 Einleitung und Motivation

Die Aufgabe von Materialflusssystemen ist das gesteuerte Transportieren von Stückgut zur Versorgung einer Fertigung (z. B. das Transportsystem einer Computerchip-Fabrik) oder innerhalb eines Logistiksystems (z. B. das Gepäcktransportsystem im Flughafen). Die ereignisdiskrete Simulation ist als Werkzeug zur Analyse der Eigenschaften und Zusammenhänge des Materialflusses in Fertigungs- und in Logistiksystemen etabliert (Banks et al. 2010). Teilweise erfolgt die Entwicklung einer Materialflusssteuerung unmittelbar im Simulationsmodell mit anschließender Übertragung in die reale Anlage. In der Emulation dient das Simulationsmodell als Ersatz realer System- und Steuerungsbestandteile (Versteegt und Verbraeck 2002; Kemper und Spieckermann 2010).

Die algorithmische Ausgestaltung der Materialflusssteuerung besitzt einen großen Einfluss auf die Leistungsfähigkeit des Systems, kann aber hinsichtlich ihrer

Korrektheit nur durch aufwändige Verfahren der Fehleranalyse und Modellevaluierung überprüft werden (vgl. Kemper und Tepper 2006). Die Schwerpunkte bisheriger Maßnahmen zur Qualitätssicherung von Materialflusssimulation liegen im Bereich Validierung und *informelle* Verifikation, wie Veröffentlichungen (z. B. Rabe et al. 2008) zeigen.

Validierung korrespondiert mit der zentralen Anforderung an ereignisdiskrete Simulation, das dynamische Systemverhalten in Raum und Zeit hinreichend realitätsnah abzubilden, um quantitative Aussagen zur Leistungsfähigkeit des Systems (Transportmengen und Transportzeiten) zu erhalten. Der gesamte Zustandsraum der Materialflusssteuerung wird dabei weder systematisch noch vollständig betrachtet. Die ereignisdiskrete Simulation ist deshalb methodisch nicht geeignet, den vollständigen Nachweis der Korrektheit der Steuerungsalgorithmen zu führen.

Der nachfolgende Artikel skizziert ein neues Verfahren zur Qualitätssicherung der Steuerung von Materialflusssystemen. Das Verfahren basiert auf dem Einsatz der formalen Verifikationstechnik Model Checking zur Systemprüfung. Model Checking realisiert eine vollständige Prüfung des Systemverhaltens durch ein automatisches Beweisverfahren. Die formale Verifikation durch Model Checking kann die ereignisdiskrete Simulation zur Analyse von Materialflusssystemen ergänzen und liefert Ergebnisse, die für eine Verbesserung der Qualität von Materialflusssteuerungen nutzbar sind.

Der Beitrag gliedert sich wie folgt: Eine verifikationsorientierte Form der Modellierung von Materialflusssystemen wird vorgestellt. An einem Beispiel wird die grundsätzliche Eignung des Konzepts für die Verifikation von Systemen realer Größenordnung dargelegt. Anschließend werden Aspekte der Abbildung von Steuerungsalgorithmen diskutiert. Der Artikel endet mit einer Zusammenfassung.

2 Model Checking

Formale Verifikation ist der Nachweis der Korrektheit des Modells eines Systems durch einen Vergleich des Modells mit einer Spezifikation. Diese Methode erfordert eine formale Beschreibung sowohl des Modells als auch der Spezifikation. Auf Basis dieser Voraussetzungen kann ein Model Checker automatisch den kompletten Zustandsraum des Modells untersuchen und die logische Korrektheit der formulierten Spezifikationen gegenüber dem Systemverhalten bestätigen oder sie mit einem Gegenbeispiel widerlegen. Das Gegenbeispiel liefert die Zustandsfolge des Systems, die zum Widerspruch gegenüber der Systemspezifikation geführt hat.

In anderen Fachgebieten findet Model Checking bereits Anwendung. Neben dem Einsatz im Schaltkreisentwurf werden Algorithmen zum Routing in Kommunikationsnetzen damit überprüft (Camara et al. 2007). Die in diesen Arbeiten beschriebenen Ansätze und Modelle sind jedoch nicht auf den Bereich Materialflusssysteme übertragbar. Eine wesentliche Ursache dafür liegt nach Einschätzung der Autoren in der Schwierigkeit und dem hohen Aufwand zur Erstellung der formal verifizierbaren Modelle und bei der Übertragung der verbalen Anforderungen in eine formale Spezifikation. Auch kann diese Methode nur bestimmte, logische Aspekte der Systemanalyse betrachten und den Einsatz der Materialflusssimulation allenfalls ergänzen. Diese Einschätzungen werden auch von Düdder et al. (2008) geteilt, die den Einsatz von Model Checking motivieren.

3 Modellierungsmethode für Materialflusssysteme

Ein Materialflusssystem wirkt als Einheit von Materialflusstechnik und Materialflusssteuerung. Aus der Anordnung und den Verbindungen verschiedener Materialflusstechnik-Komponenten ergibt sich das Layout des Systems. Solche Komponenten sind zum Beispiel Rollenförderer und Drehtische. Ausgehend vom fördertechnischen Grundverhalten der Komponenten erfolgte im ersten Schritt eine Abstraktion wesentlicher Eigenschaften in elementare Verifikationsmodelle mit zustandsorientierter Modellierung (Klotz et al. 2011).

Die für die Steuerung entscheidenden Parameter der Elemente sind:

- Kapazität (Anzahl möglicher Plätze für Transporteinheiten),
- Eingänge (Anzahl möglicher Vorgänger im Layout) und
- Ausgänge (Anzahl möglicher Nachfolger im Layout).

Technische Vorgänge ohne Ortsveränderung der Transporteinheit, z. B. Drehen auf dem Drehtisch, werden durch eine Wartephase abstrahiert. Transporteinheiten werden über ihren definierten Typ unterschieden.

Ein Element hält die Information über die sich auf ihm befindlichen Transporteinheiten und kann diese an andere Elemente übermitteln. Im Entwurf der Verhaltensmodelle wurde sichergestellt, dass die Verbindung der verifizierbaren Elementmodelle zu einem verifizierbaren Gesamtmodell des Systems führt.

Die Modellierung eines Elements schließt steuerungstechnische Bedingungen, die für die elementare Bewegung von Transporteinheiten notwendig sind, bereits ein. In der Steuerung der Bewegung der Transporteinheiten auf den Förderelementen werden zwei Ansätze unterschieden:

1. Die Ankunft einer Transporteinheit auf dem nachfolgenden Element muss von diesem quittiert werden, bevor das abgebende Element eine neue Transporteinheit aufnehmen darf. Dieses Verhalten wird typischerweise in der Palettenfördertechnik umgesetzt.
2. Die Ankunft einer Transporteinheit auf dem nachfolgenden Element muss von diesem nicht quittiert werden. Ein frei werdendes Element kann unmittelbar eine neue Transporteinheit aufnehmen. Dieses Verhalten ist für die Behälterfördertechnik typisch.

Die zustandsorientierte Modellierung für die formale Verifikation erfordert eine Diskretisierung der Werte für Entfernung und Zeit. Modellzustandsänderungen können grundsätzlich nur in diskreten Zeitschritten erfolgen. Im Modellierungsansatz wurden deshalb einige Vereinfachungen gegenüber realen Systemen getroffen.

Der Raum auf allen Förderelementen wird in Plätze gleicher Größe unterteilt, wobei ein Platz genau eine Transporteinheit aufnehmen kann. Alle Transporteinheiten besitzen die gleiche Länge. In der Bewegung wechselt somit eine Transporteinheit in einem Zustandsfolgeschritt komplett von einem Platz auf den benachbarten, sofern dieser verfügbar ist. Der abgebildete minimale Zeitschritt im Modell ist somit identisch mit der Dauer des Platzwechsels einer Transporteinheit. Die Dauer einer Wartephase im Modell beträgt immer ein ganzzahlig Vielfaches eines Zeitschritts.

Eine feinere Auflösung der Bewegung der Transporteinheiten ist innerhalb des Modellierungskonzepts durch eine weitere räumliche Unterteilung der

Förderelemente grundsätzlich möglich. Im Verifikationsmodell würde das allerdings eine Zunahme der Zustandsmenge und damit einen deutlich erhöhten Berechnungsaufwand bedeuten.

3.1 Materialflusstechnik

Für die Modellierung von Materialflusssystemen wurde eine Reihe verschiedener Fördertechnikelemente erstellt, unter anderem die folgenden:

- Quelle: Generierung einer beliebigen Sequenz von Transporteinheiten,
- Senke: Entnahme von Transporteinheiten,
- Stetigförderer: Förderstrecke mit mehreren Plätzen,
- Drehtisch: Förderelement mit mehreren Eingängen Ausgängen, eine Wartephase zur Abbildung der Arbeitsphase (Drehen) des Elements,
- Stauförderer: Förderelement mit mehreren Plätzen, die alle durch Transporteinheiten belegt werden können,
- Arbeitsstation (Bedienstation): Wartephase und Änderung des Typs der Transporteinheit möglich,
- Abweiser: Förderstrecke mit zwei Ausgängen, Push-Vorgang auf den alternativen Nachfolger mit einer Wartephase abgebildet, und
- Einschleuser: Förderstrecke mit zwei Eingängen, Wartephase zur Abbildung der Arbeitsphase (Einschleusen) des Elements konfigurierbar.

Die erstellten Elemente ermöglichen nach Erfahrung der Autoren die Modellierung kompletter Materialflusssysteme. Weitere Elemente können nach Spezifikation ihrer technischen Eigenschaften hinzugefügt werden.

Im vorgestellten Modellierungsansatz wurde von getrennten Verhaltensmodellen der Materialflusstechnik und der Materialflusssteuerung ausgegangen. Dies erlaubt eine flexible Konfiguration der Steuerung.

3.2 Routing im Gepäcktransportsystem

Stellvertretend am Beispiel des Routings in einem Gepäcktransportsystem eines internationalen Flughafens wurde die Fähigkeit des beschriebenen Verifikationsansatzes untersucht, größere Materialflusssysteme abzubilden und bestimmte Eigenschaften der Materialflusssteuerung mit Hilfe einer formalen Verifikation zu prüfen. Eine ausführliche Darstellung des Modellsystems, der Vorgehensweise und der Ergebnisse erfolgt in Klotz et al. (2012).

Neben der Modellierung der Materialflusstechnik mit den vorgestellten Elementen wurden an Verzweigungen und Zusammenführungen lokale Steuerungsentscheidungen hinzugefügt. Die möglichen Gepäcktypen an den Quellen und die erlaubten Gepäcktypen an den Zielen im System wurden spezifiziert. Der Model Checker überprüft die Korrektheit des Routing im Modell durch den automatischen Vergleich der Spezifikation mit allen in der Systemzustandsentwicklung möglichen Gepäcktypen an diesem Punkt.

Die Verifikation zeigte als Ergebnis, dass mit den hinterlegten Steuerungsregeln in allen Unterscheidungsfällen alle Gepäckstücke mit Sicherheit das richtige Ziel erreichen. Zusätzlich wurden alle vorgegebenen Zwischenstationen (z. B. Sicherheit-Screening) absolviert. Unter Anwendung eines kompositorischen Ansatzes (vorgestellt ebenda S. 10 ff.) mit der Zerlegung des Gesamtsystems in

Teilmodelle und deren getrennter Verifikation gelang der Nachweis, dass das Verfahren auch hinsichtlich Rechenzeit für eine praktische Anwendung geeignet ist. Die Durchführung der Verifikation benötigte 210 Sekunden bei 13 Teilnetzen.

Ein Simulationsmodell des Gepäcktransportsystems, erstellt mit dem Simulationsprogramm AutoMod, stand als Ausgangspunkt für Erstellung des Verifikationsmodells zu Verfügung. Das Verifikationsmodell wurde neu aufgebaut und die Steuerungslogik für das Routing manuell ergänzt.

3.3 Materialflusssteuerung

Insbesondere in der Kopplung von Simulation und Verifikation liegen das Anwendungspotential des Model Checkings und der Nutzen für die Simulation.

In Erweiterung der bisherigen Arbeiten der Autoren auf dem Gebiet werden nachfolgend die Wege einer automatisierten Informationsübertragung zwischen den verschiedenen Modellierungsarten diskutiert. Hierbei müssen die Teilbereiche Materialflusstechnik und Materialflusssteuerung getrennt betrachtet werden.

Fördertechnikkomponenten werden im Simulationssystem AutoMod ebenfalls als einzeln parametrierbare Objekte abgebildet, aus deren Verbindungen sich die Struktur des Materialflusssystems ergibt. Der ähnliche Ansatz der Modellierung der Materialflusstechnik ermöglicht die Übertragung zwischen den beiden Modellierungssystemen durch eine einfache Transformation der Beschreibungsparameter.

Die deutliche Verschiedenartigkeit der Modellierungsparadigmen beider Werkzeuge im Bereich der Materialflusssteuerung erfordert, die notwendigen und möglichen Anpassungen in beide Richtungen detaillierter zu untersuchen. Für das Ziel, eine Materialflusssteuerung aus einem Simulationsmodell (z. B. AutoMod) automatisiert in den Werkzeugen eines Model Checkers abzubilden, lassen sich drei Varianten der Transformation aufführen:

1. Anpassung der Implementierung im Simulationsmodell an die Modellierungsparadigmen des Verifikationsmodells

   Steuerungen im Simulationsmodell sind als logische Zustandsgleichungen zu formulieren und die Verwendung spezifischer Sprachkonstruktionen des Simulators zu vermeiden. Beide Anforderungen sind in AutoMod grundsätzlich erfüllbar, verursachen aber Mehraufwand in der Modellierung. Diese Steuerungen können dann zwischen den Modellen übertragen werden.

2. Übertragung von Sprachkonstruktionen des Simulationssystems in die Modellierungswelt des Verifikationsmodells

   Spezifische Sprachkonstruktionen des Simulators sind in Form von logischen Zustandsgleichungen abzubilden. Im Model Checker können dann die Formulierungen aus dem Simulationsmodell unmittelbar nachgebildet werden. Zum einen können aber möglicherweise nicht alle Sprachelemente sinnvoll per Transformation überführt werden. Auch ist die Verwendung beschränkt auf den gewählten Simulator. Diese Variante wird deshalb nicht weiter betrachtet.

3. Entwicklung einer allgemeinen Beschreibungssprache für Steuerungsalgorithmen und einer Transformation für beide Modellierungssysteme

Diese Variante hat den Vorteil, eine Schnittstelle auch für andere Modellierungssysteme zu bieten, erfordert aber den höchsten Entwicklungsaufwand. Die Algorithmen der Materialflusssteuerung als universelle parametrierbare Steuerungsbausteine abzubilden und automatisiert sowohl in ein Simulationsmodell als auch in ein Verifikationsmodell zu übertragen, ist ein Ansatzpunkt für weitere Forschungsarbeit.

Für die Abbildung der Materialflusssteuerung im Verifikationsmodell wurden die Fördertechnikelemente um Steuerungsmodule ergänzt, mit denen Steuerungslogik abgebildet werden kann. Das Steuerungsmodul kann Zustandssignale von Fördertechnikelementen empfangen und Steuersignale ausgeben.

Idee und Vorgehen der Variante 1 wird nachfolgend am Beispiel einer Zusammenführung näher erläutert. Die Steuerung einer Zusammenführung muss eine Entscheidung über die Vorfahrt treffen, wenn an beiden Eingängen Transporteinheiten (TE) warten. Eine häufig angewandte Strategie besteht in der Vorfahrt für die ältere Anforderung (Fifo-Strategie). Anhand dieser Strategie werden die Implementierungen der Steuerungen im Verifikationsmodell und im Simulationsmodell gegenübergestellt. Anschließend wird eine Variante für die Anpassung der Simulation an die Verifikation aufgezeigt.

3.3.1 Verifikationsmodell

Das Steuerungsmodul entscheidet in Auswertung der Zustände an den Ausgängen der Vorgängerelemente und des Belegungszustands der Zusammenführung. Im Algorithmus 1 wird die im Verifikationsmodell implementierte Vorfahrtsstrategie im SMV Code des Model Checkers NuSMV (Cimatti et al. 2002) im wesentlichen Ausschnitt dargestellt. Die verwendeten Variablen besitzen folgende Bedeutung:

- Pre1/2 – Zustandssignal vom Vorgänger 1/2, wenn TRUE, dann wartende TE
- Prio1 – Priorität bei Gleichzeitigkeit, wenn TRUE, dann Vorzug Vorgänger 1
- Sel1/2 – Vorfahrtentscheidung für Vorgänger 1/2
- Psel1/2 – Merker der vorangegangenen Sel1/2 Zuweisung
- Result – Steuerungsergebnis, Rückgabewert an das Fördertechnikelement

Das Modul startet mit einer Initialisierung (*init*) der internen Zustände und ermittelt in jedem Schritt die aktuelle Entscheidung neu, abgelegt in den modulinternen Variablen Sel1 und Sel2. Prio1 löst den logischen Entscheidungskonflikt bei gleichzeitiger Ankunft an Pre1 und Pre2 auf. Die Vorfahrtsentscheidung wird mit *Result* an das Zusammenführungselement übermittelt.

Algorithmus 1: *SMV Code Fifo-Strategie (Ausschnitt)*

```
MODULE merge_control_fifo(Pre1, Pre2) -- control function

ASSIGN     -- init intern variables with Boolean values
init(Psel1):= FALSE;       init(Psel2):= FALSE;
init(Prio1):= TRUE;
next(Psel1):= Sel1;        next(Psel2):= Sel2;

DEFINE
Sel1:=     -- if then else logic
case       -- evaluated in order of occurrence
```

```
  Pre1=TRUE & Pre2=FALSE : TRUE;     --     TE waits at Pre1
                                     & no TE waits at Pre2
  Pre1=TRUE & Psel1=TRUE : TRUE;     --     TE waits at Pre1
                                     & pre-booking for Pre1
  Pre1=TRUE & Psel2=FALSE & Prio1=TRUE : TRUE;
--   TE waits at Pre1 & no pre-booking Pre2, tie-prio for 1
  TRUE : FALSE;      --   all left cases set Sel1 to FALSE
esac;      -- end of case

Sel2:=     -- define Sel2 in Boolean logic
case       -- equivalent to Sel1 logic
  Pre2=TRUE & Pre1=FALSE : TRUE;
  Pre2=TRUE & Psel2=TRUE : TRUE;
  Pre2=TRUE & Psel1=FALSE & Prio1=FALSE : TRUE;
  TRUE : FALSE;
esac;      -- end of case

Result:=   -- defines new Result state
case
  Sel1=FALSE & Sel2=FALSE  : 0;
  Sel1=FALSE & Sel2=TRUE   : 2;
  Sel1=TRUE  & Sel2=FALSE  : 1;
  Sel1=TRUE  & Sel2=TRUE   : 0; -- should not happen
esac;
```

3.3.2 Simulationsmodell

Nachfolgend wird die Implementierung des vorgestellten Beispiels im Simulationssystem AutoMod mit Standardsprachelementen skizziert. Algorithmus 2 beschreibt eine einfache Steuerung der Transporteinheiten von einer Startstation über Zwischenstationen bis zur Zielstation. Die Einbindung von Zwischenstationen unmittelbar vor und nach einer Zusammenführung (*Merge*) erlaubt den Eingriff der Steuerung in den Transportablauf. Durch die Verwendung des Sprachelements *Counter* kann die Belegung einzelner Plätze (*Station*) gesteuert werden.

Ein *Counter* ist ein Zähler mit parametrierbarer, begrenzter Kapazität. Ein Überschreiten der Kapazitätsgrenze durch den Teilprozess einer Transporteinheit ist nicht möglich. Der Teilprozess wird solange in den Wartezustand versetzt, bis ein anderer Teilprozess den Zählerwert soweit verringert, dass die Erhöhung möglich wird. Die systeminterne Warteliste eines *Counters* ist nach der zeitlichen Reihenfolge der Anforderungsereignisse geordnet.

Für die Fifo-Steuerung der Zusammenführung bietet sich deshalb in AutoMod die Verwendung eines *Counters* mit Kapazität 1 an und wurde so als *counterStation* in Algorithmus 2 implementiert.

Algorithmus 2: *AutoMod Code Transport mit Zusammenführung (Ausschnitt)*

```
begin processTravel arriving procedure
   set CurrentStation to StartStation
   while (CurrentStation <> Destination) do
   begin
     set NextStation to f_GetNextStation(CurrentStation)
```

```
      if  IsMerge(NextStation)    /* wait if merge blocked */
      then increment counterStation(NextStation) by 1
      travel to NextStation       /* get on merge */
      if  IsMerge(CurrentStation) /* release blocked merge */
      then decrement counterStation(CurrentStation) by 1
      set CurrentStation to NextStation
   end
   send to processDestination
end
```

3.3.3 Simulationsmodell modifiziert

Das vorgestellte Implementierungsbeispiel zeigt, dass die Anweisungen zum Transport und zur Steuerung in der Simulation mit AutoMod häufig ineinander verzahnt sind. Diese Form ist effizient in der Simulationserstellung, aber wenig geeignet für eine Übertragung in die Verifikationsmodellierung. Die angestrebte Kopplung von Simulation und Verifikation kann damit nicht realisiert werden.

Eine Lösung besteht in der Anpassung und Erweiterung des ursprünglichen Simulationsmodells. Durch die Separierung und eine zustandsorientierte Modellierung der Steuerungsalgorithmen entsteht ein Simulationsmodell mit gleicher Funktionalität und verbesserter Transformierbarkeit.

Algorithmus 3: *AutoMod Code Zusammenführung modifiziert (Ausschnitt)*

```
begin processTravel arriving procedure
  set CurrentStation to StartStation
  while (CurrentStation <> Destination) do begin
    set NextStation to f_GetNextStation(CurrentStation)
    if  IsMerge(NextStation) then begin
        clone 1 load to processMergeControl -- start control
        wait at orderList(NextStation) -- wait for decision
    end
    travel to NextStation
     if  IsMerge(CurrentStation)
     then clone 1 load to processMergeControl
     set CurrentStation to NextStation
   end
   send to processDestination
end

begin processMergeControl arriving procedure
   set vSelect to f_MergeInSelect(CurrentMerge)
   if  vSelect <> 0
   then continue load(vSelect) from orderList(CurrentMerge)
end

begin f_MergeInSelect function
/*   evaluate merge state equivalent to algorithm 1      */

   set vMControl(CurrentMerge, PSel1)
    to vMControl(CurrentMerge, Sel1)
…
```

```
    if   vMControl(CurrentMerge, Pre1)   = true
    and  vMControl(CurrentMerge, PSel2)  = false
    and  vMControl(CurrentMerge, Prio1)  = true
    then set vMControl(CurrentMerge, Sel1) to true
…
    if   vMControl(CurrentMerge, Sel1) = true
    and  vMControl(CurrentMerge, Sel2) = false
    then set vResult to 1
…
    return vResult
end
```

Algorithmus 3 greift die Implementierung aus Algorithmus 2 auf und schlägt eine neue, erweiterte Formulierung vor. Die Transporteinheiten warten in *WaitList* auf ihre Freigabe. Die Auswahl wird separat in der Funktion *f_MergeInSelect* getroffen. Die Umsetzung der Fifo-Strategie erfolgte in zustandsorientierter Form analog zu Algorithmus 1. Die Implementierung erfolgte AutoMod Code ähnlich und wurde aus Platzgründen nur ausschnittsweise dargestellt. In gleicher Art könnten andere Vorfahrtstrategien im Algorithmus eingesetzt werden, ohne die Struktur der Implementierung zu verändern. Die Realisierung der Steuerungsentscheidung in obiger Form ausschließlich mit zustandsauswertenden logischen Funktionen ermöglicht dann deren automatisierte Übertragung in das Verifikationsmodell.

4 Zusammenfassung und Ausblick

Der Beitrag präsentiert einen Ansatz zur Verbindung der formalen Verifikation von Materialflusssystemen mit ereignisdiskreter Simulation. Die Prüfung des Systemverhaltens durch Model Checking kann die Analyse von Materialflusssteuerungen durch die ereignisdiskrete Simulation methodisch ergänzen.

Die Kombination beider Methoden bietet das Potential für eine innovative Erweiterung der derzeitigen Ansätze zur Qualitätssicherung von Materialflusssystemen. Eine direkte Kopplung von ereignisdiskreter Simulation und Model Checking bedarf einer verlustfreien Informationsübertragung zwischen den Modellen. Die Herausforderungen hierbei im Bereich der Materialflusssteuerung wurden im Artikel diskutiert und ein spezifischer Lösungsansatz für ein Simulationsprogramm wurde vorgeschlagen.

Die Entwicklung verhaltensdefinierter Steuerungselemente in einer allgemeinen Beschreibungssprache für Steuerungsalgorithmen eröffnet die Möglichkeit einer universellen Systembeschreibung, die als gemeinsamer Ausgangspunkt für verschiedene Modellanalysen dienen kann. Dieser Ansatz soll in zukünftigen Arbeiten untersucht werden. Ein weiterer Aspekt ergänzender Forschung ist der systematische Entwurf von formalen Spezifikationen für die Materialflusssteuerung aus der Analyse der gegebenen Anforderungen an das Materialflusssystem.

Danksagung

Die Arbeit wurde ermöglicht durch die Deutsche Forschungsgemeinschaft (DFG) unter den Förderkennzeichen SCHM 2689/3-1 und STR 412/4-1.

Literatur

Banks, J.; Carson II, J. S.; Nelson, B. L.; Nicol, D. M.: Discrete-event system simulation. 5. Aufl. Upper Saddle River, New Jersey: Pearson Education 2010.

Camara, D.; Loureiro, A.; Filali, F.: Methodology for formal verification of routing protocols for ad hoc wireless networks. In: Global Telecommunications Conference, 2007. IEEE, 2007, S. 705-709.

Cimatti, A.; Clarke, E.; Giunchiglia, E.; Giunchiglia, F.; Pistore, M.; Roveri, M.; Sebastiani, R.; Tacchella, A.: NuSMV 2: An OpenSource Tool for Symbolic Model Checking. In: Brinksma, v. E.; Larsen, K. G. (Hrsg.): Computer Aided Verification. Berlin, Heidelberg: Springer 2002, S. 359–364.

Clarke, E. M.; Grumberg, O.; Peled, D. A.: Model checking. Cambridge, Mass., London, MIT Press 1999.

Düdder, B.; Follert, G.; Roidl, M.: Model Checking in multiagentengesteuerten Materialflusssystemen. Technischer Bericht 817. Technische Universität Dortmund, 2008.

Grillitsch, U.; Mayer, G.: Auf dem Weg zum Standard - Virtuelle Inbetriebnahme von IT-Steuerungssystemen in der Produktionssteuerung. In: Zülch, G.; Stock, P. (Hrsg.): Integrationsaspekte der Simulation: Technik, Organisation und Personal. Karlsruhe: KIT Scientific Publishing 2010, S. 591-598.

Kemper, J.; Spieckermann, S.: Emulation von Logistik-Steuerungen in SAP-Umgebungen. In: Zülch, G.; Stock, P. (Hrsg.): Integrationsaspekte der Simulation: Technik, Organisation und Personal. Karlsruhe: KIT Scientific Publishing 2010, S. 583-590.

Kemper, P.; Tepper, C.: A Petri Net Approach to Verify and Debug Simulation Models. Dagstuhl: Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Dagstuhl Seminar Proceedings 06161, 2006.

Klotz, T.; Straube, B.; Fordran, E.; Haufe, J.; Schulze, F.; Turek, K.; Schmidt, T.: An approach to the verification of material handling systems. In: 16th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2011, Toulouse, France, 5. – 9. September 2011, S. 308-315.

Klotz, T.; Seßler, N.; Straube, B.; Fordran, E.; Turek, K.; Schönherr, J.: On the formal verification of routing in material handling systems. In: 8th IEEE International Conference on Automation Science and Engineering, CASE 2012, Seoul, 20. – 24. August 2012, S. 8-13.

Kropf, T.: Introduction to Formal Hardware Verification. Berlin, Heidelberg: Springer 1999.

Rabe, M.; Spieckermann, S.; Wenzel, S.: A new procedure model for verification and validation in production and logistics simulation. In: Mason, S. J.; Hill, R. R.; Mönch, L.; Rose, O.; Jefferson, T.; Fowler, J. W. (Hrsg.): Proceedings of the 2008 Winter Simulation Conference (WSC), Miami (USA), 7.-10. Dezember 2008, S. 1717-1726.

Versteegt, C.; Verbraeck, A.: The extended use of simulation in evaluating real-time control systems of AGVs and automated material handling systems. In: Chen, C.-H.; Yücesan, E.; Snowdon, J. L.; Charnes, J. M. (Hrsg.): Proceedings of the 2002 Winter Simulation Conference (WSC), San Diego (USA), 8.-11. Dezember 2002, S. 1659-1666.