Modellprüfung für den Entwurf von Fertigungssteuerungssystemen[1]

Stephan Flake*, Wolfgang Müller*, Ulrich Pape**, Jürgen Ruf***

* C-LAB, Universität Paderborn, Fürstenallee 11, 33102 Paderborn

** Heinz Nixdorf Institut, Universität Paderborn, Fürstenallee 11, 33102 Paderborn

*** Wilhelm-Schickard Institut, Universität Tübingen, Sand 13, 72076 Tübingen

Zusammenfassung

In diesem Artikel stellen wir die Anwendung von Modellprüfung für ein Fertigungssystem mit freifahrenden Transportfahrzeugen vor. Dabei konzentrieren wir uns darauf, den Materialfluss in einem Systemmodell auf die Einhaltung quantitativer und zeitlicher Bedingungen hin zu überprüfen. Zur Modellentwicklung setzen wir die graphische Beschreibungsmethode MFERT ein, die sich bereits mehrfach in Industrieprojekten bewährt hat. Wir präsentieren eine Abbildung von MFERT in zeitannotierte Zustandsübergangssysteme, die für ein bereits existierendes Werkzeug zur Modellprüfung als Eingabe einer Modellbeschreibung dienen. Bei der Modellprüfung wird diese Beschreibung auf die Einhaltung von Eigenschaften überprüft, die in einer temporalen Logik oder einer davon abstrahierten Form spezifiziert werden.

1 Einleitung

Bei der wachsenden Komplexität von Produktionssystemen fällt es Entwicklern schwer, den Überblick über das Verhalten des Gesamtsystems zu bewahren. Bisher wird der reibungslose Ablauf innerhalb eines Produktionssystems auf entsprechenden Modellen vorwiegend mittels Simulation überprüft. Bei dieser Art von Analyse kann aber nicht gewährleistet werden, dass alle relevanten (zeit)kritischen Situationen überprüft werden. Die formale Methode der Modellprüfung stellt eine Alternative zur Simulation dar. Modellprüfung wurde zunächst vornehmlich beim Hardwareentwurf eingesetzt, findet aber zunehmend auch in anderen Bereichen Anwendung, z. B. bei der Untersuchung von Geschäftsprozessen. In diesem Artikel wird die Anwendung von Modellprüfung nun im Zusammenhang mit fertigungsbezogener Vorgangsmodellierung und -analyse vorgestellt. Hierbei

1. Die hier vorgestellte Arbeit wurde zum Teil von der DFG im Rahmen des Schwerpunktprogramms 1064 „Integration von Techniken der Softwarespezifikation für ingenieurwissenschaftliche Anwendungen“ gefördert.

interessieren insbesondere zeitkritische Eigenschaften, die während des laufenden Betriebs vom Fertigungssystem eingehalten werden müssen. Diese Eigenschaften werden bei der Modellprüfung in der Regel als temporallogische Formeln angegeben. Ein Vorteil der Modellprüfung ist die automatisierte Überprüfung eines Modells auf Einhaltung von spezifizierten Eigenschaften. Verbesserte Strategien und Algorithmen sowie immer leistungsfähigere Computer ermöglichen inzwischen auch eine Anwendung in der Industrie. Die automatisierte Überprüfung eines Modells erlaubt auch die Ausgabe von Beispielabläufen, die zu der Verletzung einer Eigenschaft führen (Gegenbeispiele). Auf diese Weise erhalten Modellentwickler Unterstützung bei der Fehlersuche. Bei zeitkritischen Systemen können Zeitanalyseabfragen zusätzliche Informationen über die Dauer von Fertigungszeiten des vorliegenden Modells liefern.

Im folgenden Abschnitt stellen wir die diesem Artikel zugrundeliegende Fallstudie vor. Nach einführenden Kapiteln über MFERT und das formale Modell der E/A-Intervallstrukturen geben wir in Kapitel 5 exemplarisch die Abbildung einer MFERT-Beschreibung in E/A-Intervallstrukturen an. In Kapitel 6 stellen wir typische Anfragen an das Modell vor und präsentieren einige Analyseergebnisse. Kapitel 7 schließt diesen Artikel mit Anmerkungen zum Stand der Entwicklung und mit einem Ausblick ab.

2 Die Fallstudie „Produktionsautomatisierung“

Das in diesem Artikel untersuchte Szenario basiert auf einem sog. holonischen Fertigungssystem, welches als Fallstudie im Zuge der IMS Initiative eingeführt [WHS94] und später im Rahmen eines DFG-Projekts weiterentwickelt wurde [Braa99]. In diesem Zusammenhang wurde die Idee eines holonischen Fertigungssystems basierend auf der Lehre der Holarchien entwickelt, welche sich mit der Organisation von Chaos und Komplexität befasst. Ein Holon wird als autonomer, kooperativer Teil eines komplexen Ganzen verstanden, welches nur in der Summe der Einzelteile bestehen kann [Koes67]. Holonische Systeme zeichnen sich durch die Selbstähnlichkeit und Selbstkonfigurationsfähigkeit ihrer Komponenten aus, was zu einer hohen Fehlertoleranz und Stabilität des Gesamtsystems führt.

Als Grundkonfiguration in diesem Szenario sei eine werkstattorientierte Fertigungsumgebung zum maschinellen Entgraten von Gussteilen gegeben. Zum Entgraten dieser Werkstücke, z. B. Motorblöcke, Kurbelwellen oder Krümmer, werden Werkzeugmaschinen eingesetzt, z. B. eine Drei-Achsen- und eine Fünf-Achsen-Fräsmaschine, und zum nachträglichen Reinigen eine Waschmaschine. Ferner soll die Fertigungsanlage aus einem automatischen Hochregallager im Eingangsbereich, in welchem sich die zu entgratenden Werkstücke befinden, sowie einem Ausgangszwischenlager bestehen, an welches die fertiggestellten Werkstücke zum Zwecke der wei-

teren werksinternen Kommissionierung übergeben werden. Die Bearbeitungsmaschinen besitzen neben dem eigentlichen Bearbeitungsplatz für jeweils ein Werkstück einen Eingangspuffer, in welchen die zu bearbeitenden Werkstücke zwischengelagert werden können.

Der Materialfluss wird von freifahrenden, fahrerlosen Transportfahrzeugen realisiert, welche nicht als Befehlsempfänger einer übergeordneten PPS- oder Dispositionssteuerung arbeiten, sondern als eigenständige, autonome und untereinander kooperierende Holonische Transportfahrzeuge (HTFe), die mit voller Entscheidungs- und Verantwortungsbefugnis für sich selbst und damit auch für den kompletten Fertigungsablauf agieren. Zusammen bilden die Fahrzeuge das Holonische Transportsystem (HTS). Die Reihenfolge der Arbeitsstationen aus der Sicht des Materialflusses ist durch die Arbeitsschritte fest vorgegeben. Es wird weiterhin angenommen, dass pro Transportfahrt aufgrund des vereinheitlichten Werkstückträgersystems von jedem HTF immer nur ein einzelnes Werkstück übernommen und transportiert werden kann.

Im laufenden Betrieb wird das System informationstechnisch im wesentlichen durch die dynamische Interaktion mittels drahtloser Vernetzung (Broadcast) zwischen den HTFen und den Werkzeugmaschinen realisiert. Das Verhalten der Transportfahrzeuge und Werkzeugmaschinen, die im folgenden auch als Stationen bezeichnet werden, ist hierbei wie folgt skizziert.

Eine Station

1. sendet eine Anforderung für eine Auslieferung an alle HTFe
2. geht zu Schritt 1, falls kein HTF nach einer bestimmten Zeitspanne geantwortet hat
3. beendet die Verhandlungsrunde nach einer definierten Zeitspanne
4. wählt ein HTF h_i mit dem besten Angebot aus
5. geht zurück zu Schritt 1

Ein HTF

1. wartet, bis eine Auslieferungsanforderung von einer Station s_j empfangen wird
2. sendet das Angebot zur Auslieferung an s_j
3. fährt zu s_j, falls das Angebot das Beste in der Verhandlungsrunde war
4. nimmt das erste Werkstück von s_j auf, transportiert es zur nächsten Station
5. geht zurück zu Schritt 1

Physikalisch lässt sich eine Station als eine Bearbeitungseinheit mit einem Ein- und Ausgangspuffer beschränkter Größe betrachten. Die Puffer besitzen eine Be- und eine Entladeeinheit, die z. B. als einfache Förderbänder realisiert sind. Beide Einheiten müssen mit einem Sensor (z. B. einem Tastsensor) bestückt sein, der erkennt, wenn sich ein HTF an der Einheit befindet und zum Be- oder Entladen bereitsteht. In einer Ausbaustufe wäre hier auch ein realitätsnahes Handshake-Protokoll über Punkt-zu-Punkt Funkverbindung denkbar. Die zeitliche Funktion der Bearbeitungseinheit wird nur

abstrakt mit einer Bearbeitungsdauer betrachtet und umfasst keine anderen Detailgrößen wie z. B. die Rüstzeit.

3 MFERT

Die Grundlage zur Modellierung von Produktionsautomatisierungssystemen bildet bei unserem Ansatz die strukturorientierte Sprache MFERT (Modell der Fertigung) [Holt99]. MFERT stellt Konstrukte zur Modellierung von Produktionsaufgaben bereit und erhebt darüber hinaus den Anspruch, ein operables Modell zu sein, d. h., die Durchführung der Planungs- und Steuerungsaufgaben der modellierten Produktion zu ermöglichen. Dieser universelle Ansatz zur Modellierung der Fertigung ist in der Industrie akzeptiert und wurde 1997 mit dem deutschen Wissenschaftspreis Logistik ausgezeichnet [Schn96]. Die Akzeptanz von MFERT zeigt sich im erfolgreichen Einsatz in verschiedenen Industrieprojekten. Ein Beispiel ist die Modellierung der Bremsenproduktion eines international tätigen Zulieferers der Automobilindustrie [Dang98].

Ein MFERT-Modell ist auf der Basis von Fertigungselementen (F-Elemente) und Fertigungsvorgängen (F-Vorgänge) aufgebaut. Fertigungselemente repräsentieren die in der Realität vorhandenen Objekte. Sie besitzen Eigenschaften, die über Attribute beschrieben werden, und erhalten einen eigenen Identbegriff, der sich aus der Beschreibung und dem zugehörigen Zustand des Elements zusammensetzt. Mit Hilfe des Identbegriffs können einem Element die verschiedenen Stadien während der Produktion zugeordnet werden.

F-Elemente und F-Vorgänge können anhand ihrer Attribute und deren Belegungen zu Fertigungselement- bzw. Fertigungsvorgangs-Klassen zusammengefasst werden, im folgenden kurz FE- und FV-Klassen genannt. Eine FE-Klasse identifiziert eine Gruppe von F-Elementen mit gemeinsamen Merkmalen, Zeit- und Mengenrestriktionen, für die auch gemeinsame Fertigungssteuerungsmethoden angewandt werden. Dabei kann es sich um Fertigungselemente nur einer oder mehrerer Elementarten handeln. Ein Fertigungselement kann einer FE-Klasse zugeordnet werden, wenn es die Merkmale besitzt, die in der Beschreibung der FE-Klasse festgelegt wurden. Eine solche Zuordnung kann über die Zeit jedoch auch veränderlich sein. Typische Beispiele für Elemente einer FE-Klasse „Betriebsmittel“ sind Bohrmaschinen und Schraubenzieher oder für Elemente einer FE-Klasse „Materialien“ Tischbeine und -platten.

Ein Fertigungs-Vorgang (F-Vorgang) beschreibt eine Transformation über seine zugehörigen Eintritts- und Austritts-Fertigungselemente. Jeder F-Vorgang repräsentiert nur einen Vorgang und ist daher eindeutig identifizierbar. Einem F-Vorgang wird zusätzlich eine Dauer zugeordnet. Als F-Vorgänge können Veränderungen beliebiger Attribute wie Geometrie, Ort oder Status (z.B. geprüft/ ungeprüft) definiert werden. Wenn man eine FV-Klasse z.B. so definiert, dass sie die Montage von

Tischen unterschiedlicher Konstruktion umfasst, welche die Vorgangsdauer festlegt, also die Attribute des F-Vorgangs erst bei der Modellinstanziierung festgelegt werden, dann müssen diese Attribute Bestandteil der Vorgangsidentifikation sein.

Eine FV-Klasse identifiziert eine Gruppe von F-Vorgängen mit gemeinsamen Merkmalen, Zeit- und Mengenrestriktionen, für die gemeinsame Fertigungssteuerungs-Methoden angewandt werden. Dabei kann es sich um F-Vorgänge einer einzigen oder mehrerer Vorgangsarten handeln. Die FV-Klasse beschreibt den Typ eines Vorgangs und wird durch Eintritts-FE-Klassen, Austritts-FE-Klassen sowie deren Beziehungen untereinander beschrieben (einschließlich Kalender, Zeit- und Mengenrestriktionen). Beispielsweise kann eine FV-Klasse „Montage“ sowohl den Zusammenbau von runden Tischen mit 3 Beinen als auch von rechteckigen Tischen mit 4 Beinen umfassen.

Die Klassen werden von Knoten (FE-Knoten bzw. FV-Knoten) verwaltet, die einen bestimmten Zustand repräsentieren und über Kanten verbunden werden. Graphisches Symbol für einen FE-Knoten ist ein Dreieck △, während jeder FV-Knoten durch ein Viereck ▭ dargestellt wird. Eine Kante repräsentiert die Austauschbeziehung zwischen zwei Knoten bzw. zur Systemgrenze. Zu jeder Kante ist sowohl zu präzisieren, ob der Vorgänger „bringt“, „bereitstellt“ oder nur auf Anforderungen wartet und diese bedient, als auch, ob der Nachfolger „holt“, „empfängt“ oder lediglich auf Lieferungen wartet und diese vereinnahmt. Ebenfalls ist für jede Kante zu spezifizieren, unter welchen Bedingungen sie gültig ist.

Für den Aufbau komplexer hierarchischer Modelle, bei denen gegebenenfalls auch einzelne Hierarchiestufen wieder aus komplexen Modellstrukturen bestehen, existiert ein besonderer Kantentyp, die sog. Schnittstellenkante. Schnittstellenkanten erlauben die Kopplung von Modellen untereinander bzw. die Kopplung von Modellen an die reale Produktionsumgebung.

Die bisher erläuterten Modellierungskonzepte können lediglich vermitteln, wie eine Produktionsaufgabe schematisch dargestellt werden kann. Ein Beispiel hierfür ist der gemeinsame Transport von Tischbeinen, -platten und Schrauben, die anschließend zu einem Tisch montiert werden (vgl. Abbildung 1).

Zur Charakterisierung der Zustände von F-Elementen und F-Vorgängen steht das Modellierungskonstrukt Attribut zur Verfügung. Ein Attribut benennt eine Eigenschaft eines Modellelements. Es stehen ferner Modellierungskonstrukte für die Definition diskreter Zeitmodelle zur Verfügung. Für die Beschreibung von Produktionsaufgaben können innerhalb einer MFERT-Beschreibung unterschiedliche Zeitmodelle verwendet werden. Beispielsweise kann auf diese Weise modelliert werden, dass Materialien für eine Montagelinie einmalig zu Beginn einer *Schicht* bereitgestellt werden, während die montierten Endprodukte jedoch *stündlich* zum Auslieferungslager abtransportiert werden.

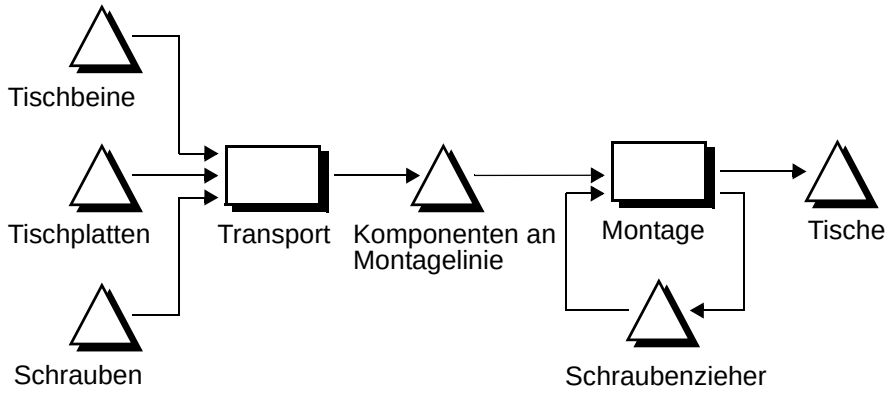

Abbildung 1. Beispiel für die Modellierung einer Produktionsaufgabe mit MFERT

Nicht beantwortet werden in der schematischen Darstellung z. B. folgende Fragen: „Was ist der aktuelle Bestand an fertiggestellten Endprodukten, Baugruppen und Teilen?“ oder „Wieviele Endprodukte sollen zukünftig produziert werden?“. Um diese Fragen beantworten zu können, durchlaufen Fertigungselemente das vorgestellte Modell einer Produktionsaufgabe. Ein bestimmter Modellzustand wird durch eine Markierung des Modells mit Fertigungselementen – die sogenannte Belegung des Modells – ausgedrückt. Die zugrundeliegende Vorstellung geht davon aus, dass die Modellknoten mit Marken gefüllt werden. Eine Marke repräsentiert je nach Knotenart entweder ein Fertigungselement oder einen Fertigungsvorgang. Damit nicht lediglich der aktuelle Modellzustand, sondern auch vergangenheitsbezogene und zukünftige geplante Zustände beschrieben werden können, werden die Marken auf sogenannten Zeit-Mengen-Leisten angeordnet, d. h., jede Marke wird einem Zeitpunkt oder Zeitraum zugeordnet.

Damit sind alle Daten im Modell abbildbar, um Produktionsautomatisierungssysteme beschreiben und im Sinne eines operablen Modells durchführen zu können. Hierzu werden den Modellknoten Funktionen zugeordnet, deren Ablaufsteuerung mittels Nachrichtenaustausch der Knoten erfolgt. Zusätzlich gibt es einen sogenannten „globalen Manager“, der den Gesamtablauf der Berechnungen auf dem Modell koordiniert und z. B. das Störungsmanagement innerhalb des Systems organisiert.

4 E/A-Intervallstrukturen

Intervallstrukturen sind Zustandsübergangssysteme mit zeitannotierten Transitionen [RuKr97]. Wir nehmen an, dass jede Intervallstruktur eine eigene Uhr zur Zeitmessung besitzt. Die Uhr wird mit jedem Zustandsübergang auf 0 zurückgesetzt. In einen anderen Zustand kann übergegangen werden, wenn der aktuelle Uhrwert mit der Verzögerungszeit an der entsprechenden ausgehenden Transition übereinstimmt. Ein Zustand muss verlassen werden, wenn die maximale Verzögerungszeit von allen ausgehenden Transitionen erreicht ist. Falls ein Zustand mehrere Nachfolgezustände hat, erfolgt eine

nicht-deterministische Auswahl. Die Verzögerungszeit wird ebenfalls nicht-deterministisch gewählt, wenn mehrere Zeiten an einer Transition angegeben sind.

Abbildung 2 zeigt eine Intervallstruktur mit zwei Zuständen auf. Von Zustand *a* darf zu den Zeitpunkten 2, 3 und 4 in den Zustand *b* gewechselt werden. Im angegebenen Beispiellauf erfolgt die Transition von *a* nach *b* zum Zeitpunkt 3, und die Zustandsuhr wird dabei gleichzeitig auf 0 zurückgesetzt.

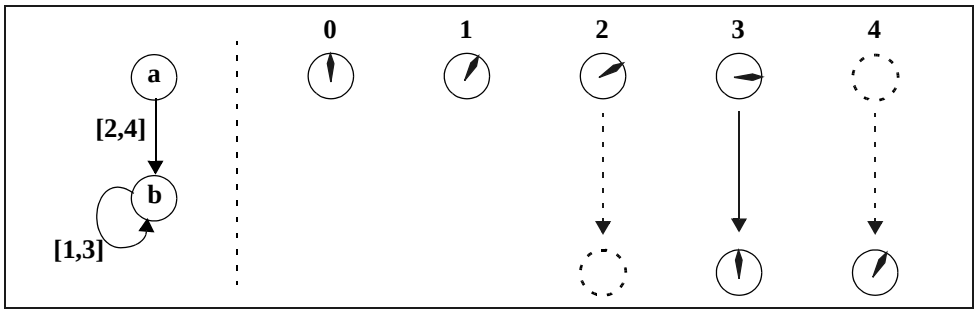

Abbildung 2. Eine Intervallstruktur mit einem Beispiellauf

Um mehrere kommunizierende Intervallstrukturen modellieren zu können, ist eine Erweiterung auf E/A-Intervallstrukturen vorgenommen worden. Diese Strukturen sind an ihren Transitionen mit Bedingungen ausgestattet, die als boolescher Ausdruck über Eingangsvariablen formuliert werden. Bei E/A-Intervallstrukturen wird eine Transition nur dann durchgeführt, wenn ihre Bedingung nicht während der zugehörigen Verzögerungszeit verletzt worden ist.

Die Entwicklung des Modellprüfers RAVEN wurde ursprünglich an der Universität Karlsruhe begonnen und wird nun an der Universität Tübingen weitergeführt [RuKr00]. RAVEN verlangt die Eingabe eines Modells als Menge von E/A-Intervallstrukturen. Im folgenden Abschnitt zeigen wir einen Ansatz auf, wie man MFERT-Beschreibungen auf E/A-Intervallstrukturen abbilden kann. Mit dieser Abbildung für MFERT-Beschreibungen kann RAVEN zur Verifikation und Analyse von Systemeigenschaften benutzt werden.

5 Abbildung von MFERT in E/A-Intervallstrukturen

Ausgehend von der Annahme, dass eine zentrale Instanz als „globaler Manager“ die Produktion steuert, beschränken wir uns zunächst auf eine synchrone Interpretation der MFERT-Beschreibungsmethode. Die Produktionsaufgabe wird in einem MFERT-Diagramm beschrieben, wie es in wesentlichen Teilen in Abbildung 3 dargestellt ist. Werkstücke gehen in die Produktion ein, werden von einem der 3 Transportfahrzeuge von der Station IN zur Maschine mill, dann (ggf. mit einem anderen HTF) zur Maschine wash und schließlich zum Ausgangslager transportiert.

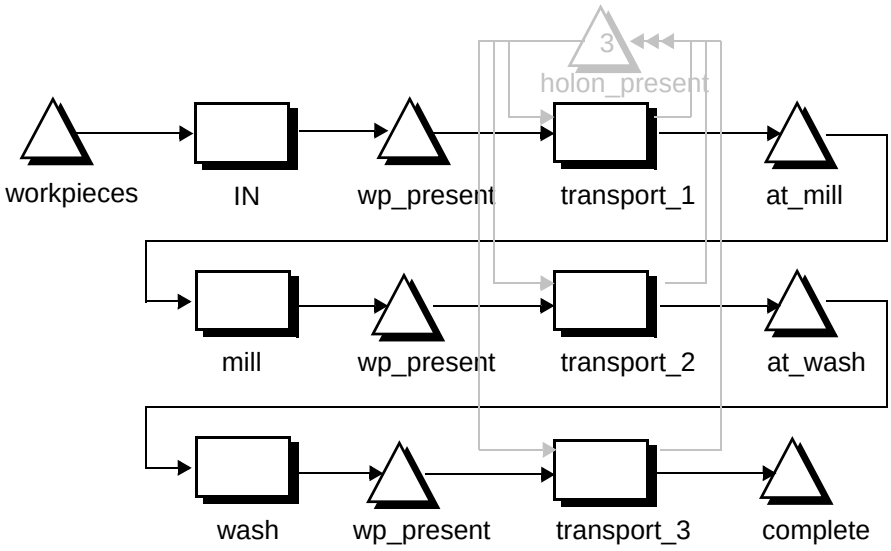

Abbildung 3. MFERT-Beschreibung der Fallstudie

Die einzelnen Vorgänge werden durch verfeinerte MFERT-Beschreibungen noch genauer modelliert, z. B. kann der Vorgang transport_1 unterteilt werden in (a) Beladen eines HTFs an der Eingangsstation, (b) HTF fährt zur Machine mill und (c) Entladen des HTFs an der Maschine mill. Wir beschränken uns hier auf den Teil (a), der das Beladen am Eingangslager IN beschreibt, und nehmen an dieser Stelle gemäß der Fallstudienbeschreibung [Braa99] an, dass immer genügend Werkstücke für die Fertigung vorhanden sind. Auf diese Weise entsteht in der Station IN ein zyklischer Ablauf zwischen dem Beladen von Transportfahrzeugen und dem Bereitstellen eines Werkstücks (vgl. Abbildung 4).

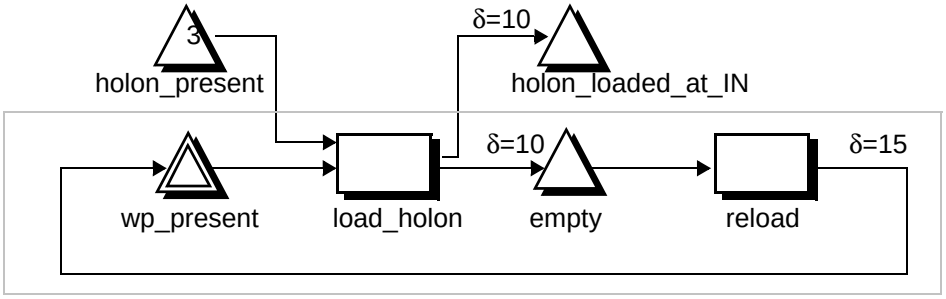

Abbildung 4. MFERT-Beschreibung zum Beladen an der Station IN

Für die Station IN muss ein Startvorgang (oder ein bei Beginn vorhandenes Fertigungselement) angegeben werden, der durch ein doppelt umrandetes Viereck (oder Dreieck) gekennzeichnet wird. In diesem Fall wird der FE-Knoten wp_present ausgewählt. Da das Beladen und Wiederbestücken in der Realität einige Zeit in Anspruch nimmt, werden an den entsprechenden Verbindungskanten in der MFERT-Beschreibung realitätskonforme Zeiten angegeben. Der FE-Knoten holon_present wird von der globalen MFERT-Beschreibung übernommen und fungiert als Schnittstellenkante zur über-

geordneten MFERT-Beschreibung, während der FE-Knoten holon_loaded_at_IN zum Modell neu hinzugefügt wird und durch eine weitere Schnittstellenkante mit dem nachfolgenden Teil (b) - HTF fährt zur Maschine mill - verbunden wird.

Übersetzung in E/A-Intervallstrukturen. Aus einer graphischen MFERT-Beschreibung wird auf folgende Weise eine E/A-Intervallstruktur gebildet, die wir in textueller Form der Eingabesprache RIL (RAVEN Input Language) des RAVEN Systems beschreiben.

Für jede MFERT-Teilbeschreibung, wie sie in Abbildung 4 beispielhaft aufgezeigt ist, wird zunächst eine eigene E/A-Intervallstruktur gebildet. Jeder Knoten wird durch einen Zustand repräsentiert, während die Kanten als Zustandsübergänge interpretiert werden. FE-Knoten, die durch Schnittstellenkanten mit der MFERT-Teilbeschreibung verbunden sind, wie z. B. holon_present, werden als *Bedingungen* interpretiert. In unserem Beispiel wird die Bedingung als boolescher Ausdruck über die 3 vorhandenen HTFe formuliert:

$$holon\_present := h1.at\_IN \;||\; h2.at\_IN \;||\; h3.at\_IN$$

Bei den Zustandsübergängen ist darauf zu achten, dass Bedingungen konsistent gehalten werden. Beispielsweise muss für die Bedingung „im Zustand wp_present und die Bedingung holon_present ist wahr“ die Alternative „im Zustand wp_present und holon_present ist falsch“ ergänzt werden, um eine gültige Beschreibung einer E/A-Intervallstruktur zu erhalten. Jedem Zustand wird genau eine Binärkombination von internen Signalvariablen zugeordnet. Dabei gibt die Anzahl *n* der Zustände die Anzahl $\lceil ld\ n \rceil$ an benötigten Signalvariablen zur Binärkodierung der Zustandsvariablen vor. Es bleibt dann nur noch eine textuelle Umsetzung vorzunehmen, wie man in Abbildung 5 nachvollziehen kann.

```
Module IN
        SIGNALS s1 s0
        INPUTS
                holon_present := h1.at_IN || h2.at_IN || h3.at_IN
        STATES
                wp_present      := !s1 & !s0
                load_holon      := !s1 &  s0
                empty           :=  s1 & !s0
                reload          :=  s1 &  s0
                holon_loaded_at_IN := empty
        INIT    wp_present
        TRANS
                wp_present  & holon_present    & load_holon'   : 1
                wp_present  & !holon_present   & wp_present'   : 1
                load_holon                     & empty'        : 10
                empty                          & reload'       : 1
                reload                         & wp_present'   : 15
END
```

Abbildung 5. RIL-Beschreibung für das Eingangslager IN

Das aufgeführte Beispiel demonstriert, dass eine Übersetzung von MFERT in E/A-Intervallstrukturen generell möglich ist. Für die Modellierung von Transportfahrten sieht die MFERT-Beschreibung deutlich komplexer aus, da eine ganze Reihe von (Zwischen-)Positionen durch verschiedene Zustände zu verwalten ist. Dies bedingt eine größere Anzahl von Signalen zur Binärdarstellung der Zustandsvariablen bei der Übersetzung in die RAVEN Input Language. So sind in der resultierenden RIL-Beschreibung unserer Fallstudie für die Verwaltung der HTF-Positionen 3*50 Zustände nötig [Ruf00].

Durch geeignete Kompositionsalgorithmen entsteht aus der Menge der auf diese Weise generierten E/A-Intervallstrukturen eine einzige, in sich geschlossene Intervallstruktur, in der die Eingangs- und Ausgabesignale in interne Zustände umgewandelt werden. Über dieser Intervallstruktur wird nun die Modellprüfung angewendet. Näheres zu den Kompositions- und Modellprüfungsalgorithmen findet man in [RuKr97].

6 Spezifikationen zur Modellprüfung

Mit dem übersetzten Modell ist die Voraussetzung für eine automatisierte Überprüfung der Eigenschaften eines Produktionssystems geschaffen. Die Eigenschaften, die das Produktionsssystem besitzen soll, müssen zur Eingabe in den Modellprüfer in temporallogischen Formeln spezifiziert werden. Wir abstrahieren aber an dieser Stelle und geben zunächst natürlichsprachliche, informelle Spezifikationen an. Vor dem Hintergrund, dass in mehreren Studien bereits existierende Eigenschaftsspezifikationen auf wiederkehrende Muster untersucht wurden, haben wir eine Menge von strukturierten, natürlichsprachlichen Spezifikationssätzen entwickelt und auf eine formale Semantik abgebildet [FMR00]. Wir gehen im Folgenden speziell auf Eigenschaften ein, die uns im Rahmen der Fallstudie interessieren. Einige Eigenschaften, die bei der Durchführung der Produktionsablaufs zu jeder Zeit gelten müssen, sind beispielsweise:

1. Die HTFe dürfen nie auf derselben Position verweilen.
2. Die Bearbeitungsmaschinen dürfen nicht länger als eine vorgegebene Zeit z leer stehen.
3. Am Ausgangslager müssen x Werkstücke innerhalb einer vorgegebenen Zeitspanne [a,b] ankommen.
4. Nach einer bestimmten Zeit z müssen bereitgestellte Werkstücke am Eingangslager abgeholt werden.

Wenn die oben genannten Eigenschaften in geeigneter Weise dem Modellprüfer übergeben werden, wird als Antwort jeweils „trifft zu“ oder „trifft nicht zu“ zurückgeliefert. Trifft eine Eigenschaft für das Modell nicht zu, wird ein Beispielablauf des Modells durch eine Liste von Zustandskonfigurationen angegeben, der zur Verletzung dieser Eigenschaft führt. Der Entwickler kann unter Zuhilfenahme dieser Informationen den Fehler beheben, indem er entweder das Modell korrigiert oder die

Eigenschaftsspezifikation umformuliert. Um diesen iterativen Prozess noch weiter zu unterstützen, kann man darüber hinaus Werte über benötigte Zeiten im laufenden Betrieb mit Zeitanalyseabfragen erhalten. Unter anderem sind folgende Anfragen sind in diesem Zusammenhang von Interesse:

1. Wann hat ein HTF x frühestens (oder spätestens) ein Werkstück zur Station y gebracht?
2. Wie lange muss ein HTF x maximal am Eingangslager warten, bis es dort ein Werkstück erhält?
3. Nach wie vielen Zeiteinheiten wird das erste Werkstück am Ausgangslager abgeliefert?
4. In welchen Abständen kann mit fertigen Werkstücken gerechnet werden?
5. Wie lange brauchen Werkstücke minimal (oder maximal) bis zur Fertigstellung in der Anlage?
6. Wie viele Werkstücke werden in der Zeit z mindestens (höchstens) fertiggestellt?
7. Wie lange steht ein HTF x längstens unbeladen - und somit ungenutzt - an einer Position?
8. Wie lange muss eine Station s mindestens (oder längstens) warten, bis ein fertiges Werkstück abgeholt wird?

Erste Untersuchungen mit einem per Hand von MFERT in E/A-Intervallstrukturen übersetzten Modell haben z. B. ergeben, dass die HTFe tatsächlich nie an derselben Position verweilen. Weitere Anfragen haben ergeben, dass nach 890 Zeiteinheiten das erste Werkstück am Ausgangslager ankommt, und dann nach jeweils 209 bis 229 Zeiteinheiten weitere Werkstücke dort angeliefert werden. Die längste Zeit, die eine Bearbeitungsmaschine stillsteht, beträgt 136 Einheiten. Weitere Details über die Untersuchungsergebnisse findet man in [Ruf00]. Eine automatische Überprüfung des Modells mit RAVEN dauerte für die Untersuchung der Kollisionsfreiheit und einiger Zeitanalyseanfragen weniger als eine Minute auf einem PC mit einem 366MHz Prozessor und 96MB Arbeitsspeicher.

7 Schlussbemerkungen

Die in diesem Artikel angegebene Abbildung ist ein erster Schritt für den Einsatz von MFERT für die automatisierte Überprüfung von Modellen durch Korrektheitseigenschaften und Analyseanfragen. Es konnten noch nicht alle MFERT-Konzepte in die Abbildung einbezogen werden; so haben wir uns zunächst auf eine synchrone Interpretation von MFERT konzentriert, was im Falle einer zentralen Steuerung jedoch keine Einschränkung in der Modellierung nach sich zieht. Es ist möglich, den Modellprüfer RAVEN zur Überprüfung von Eigenschaften an modularen Modellen mit zeitbehafteten Verbindungskanten einzusetzen. Die vorgestellten Ergebnisse sind das Resultat einer per Hand in RIL übersetzten MFERT-Beschreibung der Fallstudie. An einem graphischen Editor für MFERT und einer automatischen Übersetzung in RIL wird zur Zeit noch gearbeitet.

Für eine einprägsame Visualisierung von Beispieldurchläufen haben wir eine 3D-Animation für die Fallstudie entwickelt (vgl. Abbildung 6), die sich durch eine strukturierte Schnittstelle ansteuern lässt [BFMW00]. Bei der Modellierung der Fallstudie werden wir in weiteren Untersuchungen eine genauere Trennung zwischen physikalischem Verhalten (Bewegungen) und den Steuerungsmechanismen vornehmen. Ebenso arbeiten wir an einem erweiterten Modell der Fallstudie, das einen Wettbewerbsmechanismus zur Ersteigerung von Werkstücktransporten mit einbezieht.

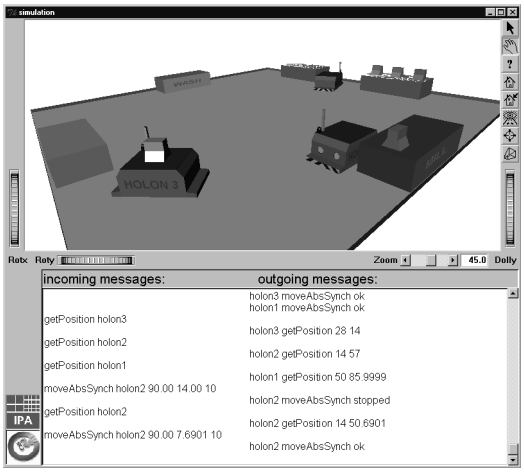

Abbildung 6. 3D-Animation

Literatur

[Braa99] Braatz, A. et al.: Referenzfallstudie Produktionstechnik v1.3: Holonischer Materialfluss, März 1999. http://www.tfs.cs.tu-berlin.de/projekte/indspec/SPP/

[BFMW00] Braatz, A.; Flake, S.; Müller, W.; Westkämper, E.: Prototyping einer Fahrzeugsteuerung in virtueller 3D-Umgebung. In: Simulation und Visualisierung 2000, Magdeburg, März 2000.

[Dang98] Dangelmaier, W.: Dynamic Production Scheduling. User Manual, Heinz Nixdorf Institut, Juli 1998.

[FMR00] Flake, S.; Müller, W.; Ruf, J.: Structured English for Model Checking Specification. In: Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen, GI/ITG/GMM Workshop, Frankfurt/M., Februar 2000.

[Holt99] Holtkamp, R.: Ein objektorientiertes Rahmenwerk zur Erstellung individueller, verteilter Fertigungslenkungssysteme. Dissertation, Heinz Nixdorf Institut, HNI-Verlagsschriftenreihe, Band 51, März 1999.

[Koes67] Koestler, A.: The Ghost in the Machine, Arkana Series, Penguin, 1967.

[Ruf00] Ruf, J.: Formal Verification of Timing Properties of a Holonic Material Transport System. Technischer Bericht, WSI-Report, Universtät Tübingen, Februar 2000.

[RuKr97] Ruf, J.; Kropf, T.: Symbolic Model Checking for a Discrete Clocked Temporal Logic with Intervals. In: Conference on Correct Hardware Design and Verification Methods (CHARME), Montreal, Kanada, Oktober 1997. IFIP WG 10.5, Chapman and Hall.

[RuKr00] Ruf, J.; Kropf, T.: Analyzing Real-Time Systems. In: Design, Automation and Test in Europe (DATE), Paris, Frankreich, März 2000. IEEE Computer Society Press.

[Schn96] Schneider, U.: Ein formales Modell und eine Klassifikation für die Fertigungssteuerung - Ein Beitrag zur Systematisierung der Fertigungssteuerung. Dissertation, Heinz Nixdorf Institut, HNI-Verlagsschriftenreihe, Band 16, 1996.

[WHS94] Westkämper, E.; Höpf, M.; Schaeffer, C.: Holonic Manufacturing Systems (HMS) - Test Case 5. In: Proceedings of Holonic Manufacturing Systems, Lake Tahoe, CA, USA, Februar 1994.