Erscheint in *9. Fachgespräch für formale Beschreibungstechniken*, Herbert Utz-Verlag, 1999

Formale Verifikation eines Reglers für Fahrbahnmarkierungsmaschinen

Peter Herrmann, Manfred Noël
Universität Dortmund, Fachbereich Informatik, D-44221 Dortmund
Email: Peter.Herrmann@cs.uni-dortmund.de, Manfred.Noel@t-online.de
Tel.: 0231/7554836, Fax: 0231/7554730

Kurzfassung

Die formale Spezifikation und Verifikation großer verteilter Systeme ist eine anspruchsvolle Aufgabe. Zum einen müssen Systembeschreibungen übersichtlich und leicht verständlich sein. Zum anderen will man interessante Systemeigenschaften trotz der Komplexität des Systemmodells mit vertretbarem Aufwand beweisen. Bei der Modellierung hybrider technischer Systeme, einer speziellen Klasse verteilter Systeme, müssen darüberhinaus Echtzeitanforderungen und kontinuierliche Flüsse berücksichtigt werden.

Wir stellen die modulare Spezifikations- und Verifikationstechnik cTLA vor, die die Spezifikation durch Komposition und die Verifikation durch Beweisstrukturierung unterstützt. Dazu verwenden wir ein reales Anwendungsbeispiel. Aufgrund einer staatlichen Vorschrift müssen Hersteller von Maschinen für die Markierung von Straßen in Deutschland zukünftig formal nachweisen, daß die erstellten Fahrbahnmarkierungen bestimmte Grenzwerte einhalten. Wir zeigen, wie man die Fahrbahnmarkierungsmaschinen spezifiziert, und skizzieren den Beweis, daß die geforderten Grenzwerte von den Maschinen eingehalten werden.

1 Einleitung

Die kompositionale Spezifikations- und Verifikationstechnik cTLA [6, 16] eignet sich gut zur formalen Beschreibung und Analyse komplexer verteilter Systeme. Da man Systemspezifikationen durch Komposition von Beschreibungen einzelner Systemkomponenten erstellt, wird die interne Struktur eines Systems in direkter und übersichtlicher Weise abgebildet. cTLA stellt sicher, daß Eigenschaften einzelner Komponenten zugleich immer auch Eigenschaften des Gesamtsystems sind, das aus ihnen gebildet ist [9]. Dadurch wird die sogenannte strukturierte Verifikation möglich, die Beweise, daß ein System interessante Systemeigenschaften erfüllt, erleichtert. Bei einem derartigen Beweis muß meist nicht die Spezifikation des Gesamtsystems sondern nur diejenige eines Subsystems berücksichtigt werden. Bei einer geeigneten Systemstruktur sind diese Subsysteme recht klein und die Beweise können einfach geführt werden.

Jedoch eignet sich die Systemstruktur realer verteilter Systeme häufig nicht für strukturierte Verifikationen, da Systemeigenschaften von der Kooperation vieler Systemkomponenten abhängen, so daß die für die Beweise verwendeten Subsysteme groß sind. cTLA

unterstützt deshalb auch constraint-orientierte Spezifikationen (vgl. [18]). Analog zum Gesamtsystem werden auch die einzelnen Systemteile kompositional spezifiziert. Sie bestehen aus Beschreibungen einzelner Constraints, die jeweils einen speziellen Aspekt der Systemkomponente modellieren. Zum Beispiel wird in einem Kommunikationsprotokoll eine Protokollinstanz durch Constraint-Spezifikationen beschrieben, die einzelne Protokollmechanismen (Sequenznummerverwaltung, Wiederholungsanforderung, ...) spezifizieren. Die Constraint-Orientierung von cTLA erleichtert die strukturierte Verifikation von Systemeigenschaften. Zum Beweis werden nur kleine Subsysteme des Gesamtsystems herangezogen, die sich aus den Spezifikationen der Constraints zusammensetzen, mit denen man diese Systemeigenschaft realisiert.

cTLA basiert auf Leslie Lamports Spezifikationstechnik Temporal Logic of Actions (TLA) [15], die um ein kompositionales Prozeßkonzept ergänzt wird (vgl. [13]). Prozesse kapseln private Zustandsgrößen ein. Aktionen definieren Transitionsklassen. Wie in der formalen Beschreibungstechnik LOTOS [12] werden Interaktionen zwischen Prozessen durch gemeinsame Aktionen modelliert. Die Aktionen besitzen Datenparameter, mit denen man den Datentransfer zwischen Prozessen beschreibt. Die formale Semantik von cTLA ist durch eine Abbildung definiert, die cTLA-Spezifikationen in semantisch äquivalente Formeln der Temporallogik TLA überführt.

Um auch Echtzeiteigenschaften und kontinuierliche Datenflüsse verteilter Systeme modellieren zu können, wurde cTLA um weitere Beschreibungsmittel ergänzt [8, 9], die sich ebenfalls an TLA anlehnen [1, 14]. Spezielle Echtzeitkonstrukte legen fest, daß Aktionen nicht über eine bestimmte Zeitspanne hinweg schaltbar sein dürfen, ohne tatsächlich geschaltet zu haben. Mit Hilfe der Aktion `CONT` kann man kontinuierliches Verhalten als Differenzen-Gleichungen modellieren. Diese Konzepte von TLA wurden an cTLA angepaßt. Insbesondere wurden Lösungen zur Wahrung der Kompositionalität, vor allem bei aktivitätsforcierenden Constraints, entwickelt.

Die Erweiterung von cTLA eignet sich auch zur Modellierung der Regelung hybrider technischer Systeme, die aufgrund der räumlichen Entfernung von diskreten Reglern, Sensoren und Aktoren spezielle verteilte Systeme sind. In [5] wurde gezeigt, daß man mit cTLA ziemlich einfach nachweisen kann, daß hybride technische Systeme wichtige Sicherheitseigenschaften erfüllen. In diesem Tagungsbeitrag soll anhand eines praktisch relevanten Beispiels eine weitere Anwendung von cTLA im Bereich der hybriden Systeme gezeigt werden. Die auf den meisten öffentlichen und privaten Straßen in Deutschland vorhandenen Fahrbahnmarkierungen werden durch spezielle Fahrbahnmarkierungsmaschinen aufgetragen. Dabei wird im allgemeinen ein Gemisch aus weißer Farbe und speziellen Reflektorperlen, die die Sichtbarkeit der Markierung bei Nacht sicherstellen, verwendet. Von besonderer Bedeutung ist die Wahl der korrekten Dicke der verwendeten Farbschicht. Bei zu hoher Schichtdicke sinken die Reflektorperlen vollständig in die Farbschicht ein und verlieren ihre reflektierende Wirkung. Bei zu geringer Schichtdicke ragen sie aus der Farbschicht heraus und werden schnell durch Fahrzeugreifen abgerieben. Die Tatsache, daß sich die Schichtdicke nur mit großem Aufwand messen läßt, führt dazu, daß einige Auftragnehmer versuchen, ihre Kosten durch geringere Schichtdicken zu reduzieren. Die öffentliche Hand wird dieses Problem durch Herausgabe neuer „Zusätzlicher Technischer Vorschriften und Richtlinien für die Markierung von Straßen“ (ZTV-M) [3] lösen. In dieser Vorschrift wird den Auftragnehmern ein elektronisches Regel- und Protokollsystem vorgeschrieben, daß die Einhaltung bestimmter Toleranzgrenzen der Schichtdicke garantiert. Ferner muß formal nachgewiesen sein, daß das Regelsystem die Toleranzgrenzen

tatsächlich einhält. Diese Nachweispflicht bildete den Ausgangspunkt für eine Diplomarbeit [17], in der für einen bestimmten Fahrbahnmarkierungsmaschinentyp ein verteiltes Regelsystem entwickelt wurde. Um die Einhaltung der Toleranzgrenzen nach der ZTV-M nachweisen zu können, wurden die Komponenten des Regelsystems und der Fahrbahnmarkierungsmaschine in cTLA spezifiziert. Durch einen cTLA-Beweis wurde schließlich die Einhaltung der ZTV-M durch das Regelsystem formal verifiziert.

Im folgenden wird zunächst auf cTLA eingegangen und die Erweiterung um Echtzeitkonstrukte und kontinuierliche Flüsse vorgestellt. Anschließend erläutern wir den Aufbau der Markierungsmaschine und des Regelsystems. Danach wird die kompositionale cTLA-Spezifikation des verteilten technischen Systems vorgestellt. Schließlich zeigen wir eine Skizze des formalen Nachweis, daß das Regelsystem die korrekte Schichtdicke einhält.

2 cTLA

Obwohl cTLA eine Notation für kanonische TLA-Formeln ist, verwendet es eine an Programmiersprachen orientierte Syntax. Spezifikationen werden durch Prozeßtypen dargestellt. Daraus werden Prozesse instantiiert, die entweder einzelne Systemkomponenten (bzw. Constraints) oder aus Komponenten zusammengesetzte Subsysteme beschreiben. Als Beispiel für einen einfachen Prozeß, der das Zustandstransitionssystem einer Komponente modelliert, ist in Abb. 1 der Prozeßtyp *Regler_Durchfluss* aufgeführt. Im Prozeßkopf ist neben dem Prozeßnamen der generische Parameter `k` angegeben, mit dem man spezielle Ausprägungen der Prozeßinstanzen spezifiziert. Der Prozeßrumpf beschreibt das Zustandstransitionssystem der modellierten Komponente. Im Beispiel wird der Zustandsraum durch die Variable `f` aufgespannt. Die Bedingung `INIT` modelliert die Initialzustände. Transitionsmengen werden durch Aktionen beschrieben. Eine Aktion ist ein Prädikat über Aktionsparameter (z.B. `hf`) sowie über Paaren von Zuständen vor bzw. nach Aktionsausführung. Dabei werden Variablen, die einen Zustand nach Aktionsausführung ausdrücken, durch das spezielle $'$-Symbol gekennzeichnet (z.B. `f'`). Die Disjunktion der Aktionen im Prozeßtyp beschreibt die Menge aller Transitionen der Komponente. Zusätzlich sind immer Stotterschritte erlaubt, in denen sich der Zustand der Komponente nicht ändert.

Der Prozeßtyp *Regler_Durchfluss* spezifiziert eine Safetyeigenschaft (vgl. [2]). Livenesseigenschaften, die den Systemfortschritt beschreiben, werden durch spezielle Fairness-

Komponente des Reglers für die Markierungsmaschine, der Durchfluss an Farbe beruecksichtigt
`PROCESS Regler_Durchfluss (k : Real)`
k : Faktor zwischen Durchfluss und Stellventil-Einstellung
`VARIABLES f : real;` *Aktueller Durchfluss*
`INIT` $\triangleq$ `f = 0;`
`ACTIONS`
`LESE_DURCHFLUSS (hf : Real)` $\triangleq$ *Lese Durchfluss-Sensor (Parameter hf*
`f' = hf;`
`SETZE_STELLVENTIL (og : Real)` $\triangleq$ *Berechne Stellung des Stellventils (Parameter og)*
`og = k · f ∧ f' = f;`
`END`

Abbildung 1: cTLA-Prozeßtyp *Regler_Durchfluss*

anforderungen an Aktionen dargestellt. Schwache Fairness drückt aus, daß eine Aktion nicht ständig schaltbar sein darf, ohne irgendwann tatsächlich zu schalten. Starke Fairness verlangt darüberhinaus, daß eine Aktion schaltet, wenn sie zwar unendlich oft schaltbar ist, es dazwischen aber Zustände gibt, in der sie nicht geschaltet werden darf. Um die Konsistenz von Prozeßkompositionen sicherzustellen, hängt die Fairness in cTLA im Gegensatz zu TLA auch von der Umgebung eines Prozesses ab. Eine Prozeßaktion muß nur schalten, wenn es genügend Zustände gibt, in denen sowohl die Schaltbedingung der Aktion gilt als auch ihr Schalten von der Prozeßumgebung toleriert wird. In cTLA-Prozeßtypen wird Fairness durch Ausdrücke der Form `WF: LESE_DURCHFLUSS` (schwache Fairness) bzw. `SF: SETZE_STELLVENTIL` (starke Fairness) dargestellt.

Echtzeit wird durch eine spezielle Zustandsgröße `now` repräsentiert, die von einer Aktion `tick` in sehr kleinen Schritten erhöht wird (vgl. [1]). `now` kann in allen Prozessen gelesen werden. Mit zusätzlichen cTLA-Konstrukten kann man minimale Wartezeiten und maximale Reaktionszeiten von Prozeßaktionen definieren. Analog zu den Unterschieden zwischen starker und schwacher Fairness werden dabei verletzliche oder persistente Zeitspannen festgelegt. Eine verletzliche Zeitspanne berücksichtigt nur den Zeitraum, in dem eine Aktion ununterbrochen schaltbar ist. Dagegen können persistente Zeitspannen von Perioden unterbrochen werden, in denen die Aktion nicht geschaltet werden darf. Die Notation entspricht derjenigen von Fairness-Anforderungen. Nach den Schlüsselwörtern `V` (verletzlich) oder `P` (persistent) sowie `MIN TIME` oder `MAX TIME` wird der Aktionsname und die Zeitspanne angegeben. Zum Beispiel zeigt `V MAX TIME: SETZE_STELLVENTIL 0.25`, daß die Aktion `SETZE_STELLVENTIL` geschaltet haben muß, bevor sie 0,25 Zeiteinheiten ununterbrochen schaltbar war.

Kontinuierliches Verhalten in Prozessen wird durch die spezielle zeitbehaftete Aktion `CONT` modelliert, in der Differenzen-Gleichungen über reellwertige Variablen aufgelistet sind. Kontinuierliche Flüsse werden durch eine Folge von `CONT`-Schritten approximiert, die jeweils sehr kleine Zeitschritte beschreiben (vgl. [14]). Durch spezielle Aktionsparameter modelliert man kontinuierliche Ein- und Ausgaben. Die `CONT`-Aktionen und die `tick`-Aktion werden stets gleichzeitig geschaltet. Als Beispiel ist in Abb. 2 der Prozeß *Schichtdicke* aufgeführt, der die Dicke der auf die Straße aufgetragenen Farbschicht spezifiziert. Die aktuelle Schichtdicke wird durch die Variable `sd` beschrieben. Die Eingabeparameter `vi` und `fi` der Aktion `CONT` modellieren die Geschwindigkeit der Fahrbahnmaschine bzw. den Ausstoß an Farbe, von denen die Schichtdicke abhängt. Der aktuelle Wert von `sd` wird durch die Differenzen-Gleichung `sd' = fi / (vi · (now'-now) · sb)` spezifiziert. Da-

```
Auf Fahrbahn aufgetragene Schichtdicke
PROCESS Schichtdicke (sb : Real)  sb : Strichbreite
VARIABLES sd : real;  Aktuelle Schichtdicke
INIT ≙ sd = 0;
ACTIONS
  CONT (INPUT vi, fi : Real;
        OUTPUT og : Real) ≙  Kontinuierliches Verhalten
    sdo = sd ∧
    sd´ = fi / (vi · (now´-now) · sb);
END
```

Abbildung 2: cTLA-Prozeßtyp *Schichtdicke*

```
System aus Regler, Fahrbahnmarkierungsautomat und Strecke
PROCESS Regelsystem (sb : Real)
PROCESSES
  V   : Geschwindigkeit (-7, 7, 0.35, 0.7); Geschwindigkeit der Maschine
  F   : Durchfluss (100000,1500);   Durchfluss an Farbe
  SD  : Schichtdicke (0.3);         Dicke der aufgetragenen Farbschicht
  RD  : Regler_Durchfluss (0.377); Regler abhaengig vom Durchfluss
  RDL : Zeitschranke (0.02);        Echtzeit-Anforderung an LESE_DURCHFLUSS in RD
  RDS : Zeitschranke (0.02);        Echtzeit-Anforderung an SETZE_STELLVENTIL in RD
  STV : Stellventil;                Stellventil
  SF  : Sensor_Durchfluss;          Sensor fuer den Farbdurchfluss
  ...;
ACTIONS
  CONT (OUTPUT ... : Real) ≙ Kontinuierliches Verhalten
    V.CONT (; vo) ∧ F.CONT (; fo) ∧ SD.CONT (vo, fo; sdo) ∧
    STV.CONT (; ogo) ∧ SF.CONT (fo ;) ∧ ...;
  LESE_DURCHFLUSS (df : Real) ≙ Lese Durchfluss-Sensor
    RD.LESE_DURCHFLUSS (df) ∧ RDL.SIGNAL ∧  SF.AUSGABE (df) ∧ ...;
  SETZE_STELLVENTIL (og : Real) ≙ Stelle das Stellventil
    RD.SETZE_STELLVENTIL (og) ∧ RDS.SIGNAL ∧ STV.OEFFNE (og) ∧ ...;
  ...;
END
```

Abbildung 3: cTLA-Systemtyp *Regelsystem*

bei beschriebt `(now'-now)` den von `CONT` modellierten Zeitschritt. Der Ausgabeparameter `sdo` exportiert die aktuelle Schichtdicke an andere Prozesse.

System- und Subsystemspezifikationen werden aus Prozeßspezifikationen komponiert. Der Zustand des Systems entspricht dem Vektor der Prozeßzustände. Eine Systemaktion wird als Konjunktion von Aktionen der Einzelprozesse spezifiziert, die somit simultan ausgeführt werden. Dabei schaltet jeder Prozeß entweder genau eine Prozeßaktion oder einen Stotterschritt, so daß Nebenläufigkeit durch Interleaving modelliert wird. Die Kopplung von Prozessen erfolgt wie in LOTOS [12] durch gemeinsam geschaltete Aktionen. Datenaustausch zwischen den Prozessen spezifiziert man mit Hilfe der Aktionsparameter, da gleichnamige Parameter aller in einer Systemaktion beteiligten Prozeßaktionen denselben Wert haben müssen. Der Prozeß *Regelsystem* in Abb. 3 ist ein Beispiel für eine cTLA-Systemspezifikation. Im Feld `PROCESSES` werden die Prozesse des Systems mit den aktuellen Belegungen der Prozeßparameter beschrieben. Die Kopplung der Prozeßaktionen zu Systemaktionen ist in Feld `ACTIONS` spezifiziert.

3 Markierungsmaschine und Regelsystem

Für die Herstellung von Längsmarkierungen (Rand- und Leitlinien) werden, wie bereits in der Einleitung beschrieben, selbstfahrende Markierungsmaschinen eingesetzt. Diese Maschinen haben einen oder zwei luftdruckbeaufschlagte Materialbehälter mit einem Fassungsvermögen von jeweils 500 bis 1000 l. Über ein Schlauchsystem wird das Material an eine oder zwei Spritzpistolen geleitet, die es — häufig mit zusätzlicher Zerstäuberluft — auf die Fahrbahnoberfläche auftragen. Nachdem der Kompressor den notwendigen

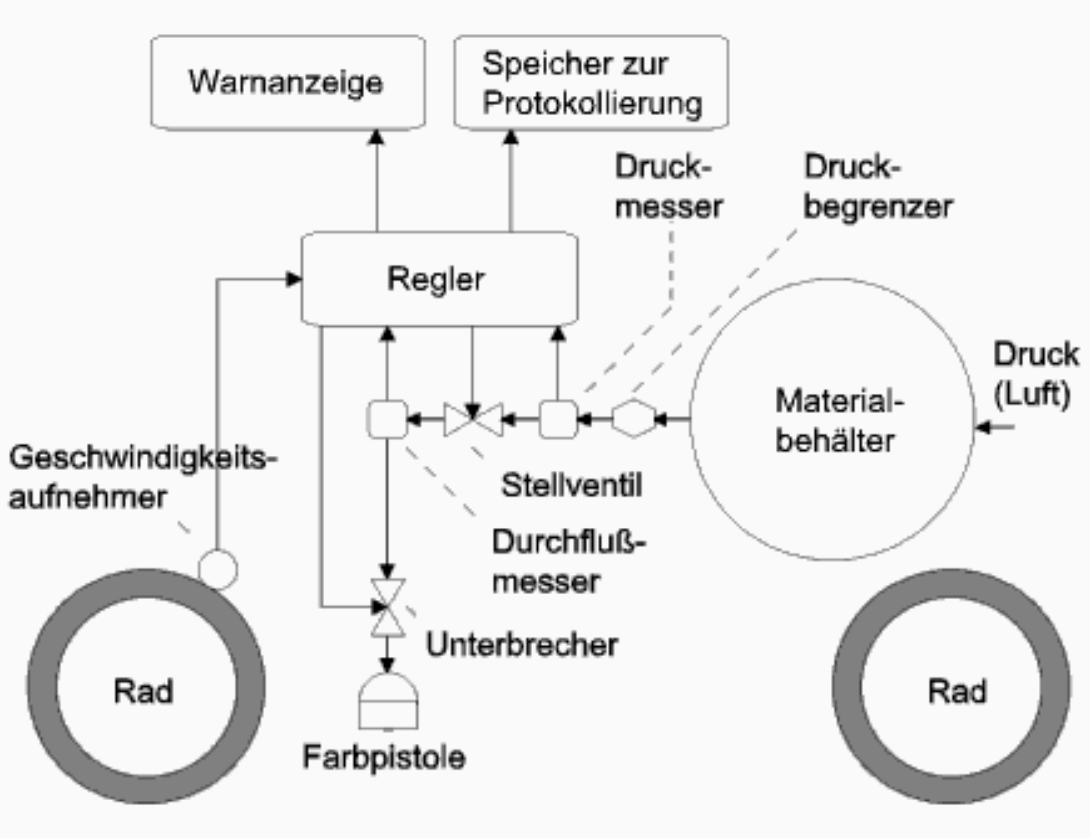

Abbildung 4: Das geregelte System

Druck im Behälter erzeugt hat, fährt der Fahrer mit einer Geschwindigkeit von 5 bis 25 km/h möglichst exakt entweder über eine schon vorhandene Markierung oder, bei Neubaustrecken, über eine speziell dafür angelegte Vormarkierung. Die mögliche Arbeitsgeschwindigkeit steht in direktem Zusammenhang mit der Strichbreite, der Schichtdicke und dem verwendeten Material sowie dem Straßenverlauf. Bezüglich der Ansteuerung der Pistolen gibt es sowohl die Möglichkeit eines automatischen Zyklus, bei dem sich nach zuvor eingestellten Längen von Strich und Lücke die Pistolen automatisch öffnen und schließen, als auch die manuelle Variante, bei der der Maschinenfahrer diese Funktion übernimmt. Die Schichtdicke, die nun im Mittelpunkt steht, stellt also die zu regelnde Größe dar und ist vor allem von den Störgrößen Geschwindigkeit und Druck bzw. Durchfluß abhängig.

Die Regeleinrichtung ist in Abb. 4 abgebildet. Zunächst wird der Materialfluß vom Behälter bis zur Spritzpistole beschrieben. Ein erheblicher Störfaktor des bestehenden Systems im Hinblick auf die Regelung ist die starke Druckschwankung im Bereich des Materialbehälters. Um diese Schwankung abzumildern, wird direkt zwischen dem Materialbehälter und dem Stellventil ein Druckbegrenzer eingebaut. Im weiteren Verlauf des Materialflusses wird der Druck gemessen, um die Pistole nur bei genügend hohem Druck zu öffnen. Das Material fließt weiter durch ein lineares Stellventil und einen Durchflußmesser, wobei vor der Farbpistole noch ein Unterbrecher angebracht ist, der einen direkten Stillstand des Materialflusses ermöglicht.

Der Regler erhält als Eingabe den aktuellen Druck, Durchfluß sowie die Geschwindigkeit des Fahrzeugs. Unter Verwendung der Daten über den Druck wird der Unterbrecher geregelt, während das Stellventil vom Durchfluß und von der Geschwindigkeit abhängt.

Als Vorgaben erhält der Regler den maximalen erlaubten Materialdurchfluß sowie die minimale und die maximale Geschwindigkeit. Mit diesen Daten soll außer der Schichtdickenregelung noch folgendes gewährleistet werden: Der Maschinenfahrer soll auf der einen Seite nicht die Möglichkeit haben, bei zu geringer Geschwindigkeit die Pistole zu öffnen und zu markieren, da die notwendige Randschärfe erst ab einer bestimmten Geschwindigkeit erreicht werden kann. Auf der anderen Seite soll auf einem Display eine Warnung erscheinen, wenn die Geschwindigkeit so groß wird, daß es technisch nicht möglich ist, einen dementsprechenden Durchfluß zu erhalten. Nach einer kurzen Toleranzzeit wird die Pistole dann automatisch geschlossen.

4 Spezifikation

Der Regler für Fahrbahnmarkierungsautomaten wird durch den cTLA-Prozeßtyp *Regelsystem* (siehe Abb. 3) modelliert. *Regelsystem* enthält Spezifikationen einzelner Systemkomponenten, mit denen man die Regelstrecke, die Aktoren, die Sensoren und den Regler beschreibt. Die Regelstrecke wird durch vier cTLA-Prozesse modelliert, die die Constraints Geschwindigkeit, Druck, Durchfluß und Schichtdicke spezifizieren. Da die Regelstrecke dem kontinuierlichen Anteil des Systems entspricht, verwenden diese Prozesse `CONT` als einzige Aktion. Die Prozeßtypen *Geschwindigkeit* und *Druck* werden mit jeweils vier festen Prozeßparametern für obere und untere Grenzwerte sowie Grenzwertänderungen versehen und geben die aktuelle Geschwindigkeit des Fahrzeugs bzw. den Druck im Leitungssystem an. Der Prozeßtyp *Durchfluss* hat den aktuellen Druck, den prozentualen Öffnungsgrad des Stellventils sowie die Position des Unterbrechers als Eingabegrößen der Aktion `CONT`. Daraus wird der aktuelle Durchfluß an Farbe unter Anwendung der Energiegleichung von Bernoulli berechnet. Der Prozeßtyp *Schichtdicke* (siehe Abb. 2) spezifiziert die Dicke der auf die Fahrbahn aufgetragenen Farbschicht. Sie hängt von der als Prozeßparameter spezifizierten Strichbreite sowie von den `CONT`-Eingabegrößen Geschwindigkeit und Durchfluß ab.

Der Regler beeinflußt die Regelstrecke durch das Stellventil und den Unterbrecher. Diese Aktoren werden durch zwei cTLA-Prozesse modelliert, die neben `CONT` weitere Aktionen enthalten, mit denen man die diskreten Steuerkommandos des Reglers spezifiziert. Die Sensoren des Regelsystems messen die Geschwindigkeit, den Druck sowie den Durchfluß und enthalten neben `CONT` Aktionen, die die Übergabe der gemessenen Werte an den Regler beschreiben.

Wie in Kap. 3 schon erläutert wurde, erfüllt der diskrete Regler zwei Aufgaben. Zum einen wird der Unterbrecher bei zu geringem Druck geschlossen, was durch den Prozeßtyp *Regler_Druck* modelliert wird. Zum anderen steuert der Regler das Steuerventil in Abhängigkeit vom Durchfluß der Farbe und der Geschwindigkeit des Markierungsautomaten. Dieser Teil wird durch zwei cTLA-Prozesse spezifiziert. Der in Abb. 1 aufgeführte Prozeßtyp *Regler_Durchfluss* spezifiziert die Regelung des Stellventils nur unter Berücksichtigung des Durchfluß. Der aktuelle Durchfluß wird dabei zunächst vom Durchflußsensor abgelesen (Aktion `LESE_DURCHFLUSS`). Daraus errechnet der Regler die neue Ventilstellung und überträgt dem Ventil ein entsprechenes Stellkommando (Aktion `SETZE_STELLVENTIL`). Mit dem cTLA-Prozeßtyp *Regler_Geschwindigkeit* modelliert man die Regelung des Stellventils abhängig von der Geschwindigkeit des Fahrzeugs. Die Echtzeitanforderungen an den Regler spezifiziert man durch fünf weitere Prozesse, die alle vom

Prozeßtyp *Zeitschranke* instantiiert werden. *Zeitschranke* enthält eine Aktion `SIGNAL`, die nach spätestens 20 ms persistent schalten muß. Die fünf Aktionen der Regler-Prozesse, die das Lesen von Sensoren und das Stellen von Aktoren spezifizieren, werden jeweils mit einer der fünf Aktionen `SIGNAL` gekoppelt und übernehmen damit deren Echtzeitanforderung.

5 Verifikation

Im folgenden skizzieren wir den Beweis, daß das in Kap. 4 spezifizierte System zur Regelung von Fahrbahnmarkierungsautomaten die Farbe mit einer Schichtdicke aufträgt, die von der geforderten Norm um höchstens 10 % abweicht und somit die von der ZTV-M festgelegte Toleranzgrenze [3] einhält. Diese Eigenschaft kann in TLA durch die Formel

$$I \triangleq (0,9 \cdot vsd \leq SD.sd \leq 1,1 \cdot vsd) \vee (UB.ou = 0)$$

beschrieben werden. Die Formel drückt aus, daß die aktuelle Schichtdicke (Variable `sd` von Prozeß `SD`) entweder um höchstens 10 % von der vorgegebenen Schichtdicke `vsd` abweicht, oder daß der Unterbrecher geschlossen ist. Wir müssen beweisen, daß I eine Invariante des geregelten Systems ist. Das entspricht in TLA der Formel

$$Regelsystem \Rightarrow \Box I$$

Invariantenbeweise werden in zwei Schritten durchgeführt. Zum einen beweist man, daß I zum Systemstart gilt. Es muß aus den Initialbedingungen der in *Regelsystem* gekoppelten Prozesse gefolgert werden können. Zum anderen wird verifiziert, daß I über die Systemaktionen von *Regelsystem* stabil ist. Leider ist I zu schwach, um direkt bewiesen zu werden. Man muß eine stärkere Invariante I_s auswählen, für die ein direkter Invariantenbeweis möglich ist. In unserem Fall ist I_s eine Konjunktion über I und einigen weiteren TLA-Formeln, mit denen man die folgenden Eigenschaften des geregelten Systems beschreibt:

- Der Unterbrecher darf nur geöffnet werden, wenn die Dicke der anschließend aufgetragenen Farbe innerhalb der Toleranzgrenzen liegt.
- Der Unterbrecher wird geschlossen, bevor die Schichtdicke die erlaubten Toleranzgrenzen über- bzw. unterschreitet.

Zum Beweis von I_s muß nicht das Gesamtsystem *Regelsystem* sondern ein kleineres Subsystem verwendet werden, das nur aus den Prozessen der Regelstrecke, dem Unterbrecher, den Sensoren sowie den Reglerkomponenten, die den Unterbrecher steuern, besteht. Die Formel

$$Subsystem \Rightarrow \Box I_s$$

wird in Form eines Invariantenbeweis verifiziert. Aufgrund der Definition von I_s folgt daraus $Subsystem \Rightarrow I$.

Nach der Definition von cTLA ist eine Eigenschaft eines Subsystems zugleich eine Eigenschaft jedes Gesamtsystems, in dem das Subsystem enthalten ist. Dazu dürfen allerdings die Aktionen des Subsystems, die Fairnessanforderungen oder Echtzeitanforderungen bezüglich maximaler Reaktionszeiten besitzen, nicht durch Aktionen anderer Prozesse des Gesamtsystems blockiert werden. In unserem Beispiel ist dieser Beweis trivial, da alle Prozesse von *Regelsystem*, die nicht in *Subsystem* enthalten sind, an den Aktionen von

Subsystem mit einem Stotterschritt oder mit einer Aktion, die immer schalten darf, gekoppelt sind. Da somit auch $Regelsystem \Rightarrow Subsystem$ gilt, ist I eine Invariante von $Regelsystem$ und wir haben sichergestellt, daß unser System die Anforderungen der ZTV-M erfüllt. Ein weiterer Beweis hat gezeigt, daß die Regelung des Stellventils ausreicht, um Abschaltungen des Unterbrechers wegen Verletzung der Toleranzgrenzen im normalen Betriebsablauf zu vermeiden.

6 Schlußbetrachtungen

Im Beitrag wurde anhand eines realen Anwendungsbeispiels gezeigt, daß man mit cTLA übersichtliche Spezifikationen hybrider verteilte Systeme erstellen kann. Für die Verifikation der Systemeigenschaften konnten kleinere Subsysteme verwendet werden, so daß der Beweis vereinfacht wurde. Das Beispielsystem wurde in einer Woche spezifiziert und verifiziert.

cTLA unterstützt auch die Wiederverwendbarkeit von Spezifikationen und Beweisen. Sogenannte Spezifikationsframeworks [7, 11] enthalten Bibliotheken von cTLA-Prozeßtypen und von Theoremen. Ein System wird spezifiziert, indem man geeignete Prozeßtypen des Frameworks instantiiert und miteinander komponiert. Die Theoreme des Frameworks bringen zum Ausdruck, daß eine Systemeigenschaft durch ein aus Spezifikationsbausteinen zusammengesetztes Subsystem realisiert wird. Da die Theoreme bereits bewiesen sind, reduziert sich die Verifikation von Systemeigenschaften auf einige einfache Konsistenzüberprüfungen. Darüberhinaus unterstützt ein Werkzeug [4] die Auswahl geeigneter Theoreme und die Durchführung der Konsistenzüberprüfungsschritte. Der Framework-Ansatz wurde mit Erfolg im Bereich der Kommunikationsprotokolle eingesetzt (vgl. [10]). Außerdem existiert ein Framework für Sicherheitsüberprüfungen technischer Anlagen, das im WWW unter `http://ls4-www.cs.uni-dortmund.de/RVS/P-HYSYS` verfügbar ist. Die hier vorgestellte Arbeit dient als Grundlage für ein weiteres Framework, das die Spezifikation und Verifikation von Regelungen technischer Systeme vereinfachen soll.

Literatur

[1] M. Abadi und L. Lamport. An old-fashioned recipe for real time. In J. W. de Bakker, C. Huizing, W. P. de Roever und G. Rozenberg (Hrsg.), *Real-Time: Theory in Practice*, LNCS 600. Springer-Verlag, 1991.

[2] B. Alpern und F. B. Schneider. Defining liveness. *Information Processing Letters*, 21:181–185, 1985.

[3] Fachgremium ZTV-M. *Zusätzliche Technische Vorschriften und Richtlinien für die Markierung von Straßen*, 1999.

[4] P. Herrmann, O. Drögehorn, W. Geisselhardt und H. Krumm. Tool-supported formal verification of highspeed transfer protocol designs. In *Proceedings of the 7th International Conference on Telecommunication Systems — Modelling and Analysis*, S. 531–541, Nashville, März 1999. ATSMA, IFIP.

[5] P. Herrmann, G. Graw und H. Krumm. Compositional Specification and Structured Verification of Hybrid Systems in cTLA. In *Proceedings of the 1st IEEE International Symposium*

on Object-oriented Real-time distributed Computing (ISORC98), S. 335–340, Kyoto, April 1998. IEEE Computer Society Press.

[6] P. Herrmann und H. Krumm. Compositional Specification and Verification of High-Speed Transfer Protocols. In S. T. Vuong und S. T. Chanson (Hrsg.), *Protocol Specification, Testing, and Verification XIV*, S. 339–346, Vancouver, 1994. IFIP, Chapman & Hall.

[7] P. Herrmann und H. Krumm. Re-Usable Verification Elements for High-Speed Transfer Protocol Configurations. In P. Dembiński und M. Średniawa (Hrsg.), *Protocol Specification, Testing, and Verification XV*, S. 171–186, Warschau, 1995. IFIP, Chapman & Hall.

[8] P. Herrmann und H. Krumm. Kompositionale Constraints hybrider Systeme. In E. Schnieder und D. Abel (Hrsg.), *Entwurf komplexer Automatisierungssysteme*, S. 243–264, Braunschweig, 1997.

[9] P. Herrmann und H. Krumm. Specification of Hybrid Systems in cTLA+. In *Proceedings of the 5th International Workshop on Parallel & Distributed Real-Time Systems (WPDRTS'97)*, S. 212–216, Genf, 1997. IEEE Computer Society Press.

[10] P. Herrmann und H. Krumm. Modular Specification and Verification of XTP. *Telecommunication Systems*, 9(2):207–221, 1998.

[11] P. Herrmann und H. Krumm. Protokollspezifikation und -verifikation mit dem Transferprotokoll-Framework. *Praxis der Informationsverarbeitung und Kommunikation (PIK)*, 21(2):79–88, 1998.

[12] ISO. *LOTOS: Language for the temporal ordering specification of observational behaviour*, International Standard ISO/IS 8807 Edition, 1989.

[13] R. Kurki-Suonio. Hybrid Models with Fairness and Distributed Clocks. In R. L. Grossmann, A. Nerode, A. Ravn und H. Rischel (Hrsg.), *Proceedings of a Workshop on Theory of Hybrid Systems*, LNCS 736, S. 103–120, Lyngby, Dänemark, Okt. 1993. Springer-Verlag.

[14] L. Lamport. Hybrid Systems in TLA^+. In R. L. Grossmann, A. Nerode, A. Ravn und H. Rischel (Hrsg.), *Proceedings of a Workshop on Theory of Hybrid Systems*, LNCS 736, S. 77–102, Lyngby, Dänemark, Okt. 1993. Springer-Verlag.

[15] L. Lamport. The Temporal Logic of Actions. *ACM Transactions on Programming Languages and Systems*, 16(3):872–923, Mai 1994.

[16] A. Mester und H. Krumm. Composition and Refinement Mapping based Construction of Distributed Applications. In *Proceedings of the Workshop on Tools and Algorithms for the Construction and Analysis of Systems*, Aarhus, Dänemark, 1995. BRICS.

[17] M. Noël. Korrektheitssichernder Entwurf eines Reglers für Fahrbahnmarkierungsautomaten. Diplomarbeit, Universität Dortmund, Informatik IV, D-44221 Dortmund, 1998.

[18] C. A. Vissers, G. Scollo und M. van Sinderen. Architecture and specification style in formal descriptions of distributed systems. In S. Agarwal und K. Sabnani (Hrsg.), *Protocol Specification, Testing and Verification VIII*, S. 189–204, Elsevier, 1988. IFIP.