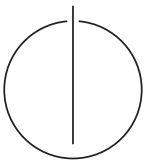
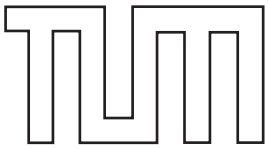


TECHNISCHE UNIVERSITÄT MÜNCHEN
FAKULTÄT FÜR INFORMATIK

Software & Systems Engineering
Prof. Dr. Dr. h.c. Manfred Broy



SPES 2020 Deliverable D1.1.B-1

Modellierungstheorie

Liste an Feedback und Änderungswünschen der

SPES-Projektpartner

Author: Alexander Harhurin
Judith Thyssen
Version: 1.0
Date: December 31, 2009
Status: Released

Technische Universität München - Fakultät für Informatik - Boltzmannstr. 3 - 85748 Garching

1 Einführung

Dieses Dokument fasst die Kommentare der SPES-Projektpartner zu der Modellierungstheorie, wie vom Lehrstuhl Prof. Broy im Rahmen des Deliverables D1.1.A-1 [HHR09] vorgeschlagen, zusammen. Eine Auswahl und Bewertung der Änderungswünsche ist nicht Gegenstand dieses Dokuments, sondern wird Inhalt des folgenden Deliverables D1.1.B-2 sein.

Die wichtigsten Feedback-Punkte und Änderungswünsche lassen sich wie folgt kurz zusammenfassen:

- *Annahme eines globalen, diskreten Zeitmodells.* Von allen Feedback-Partnern wurde das verwendete Zeitmodell, das auf einer globalen, diskreten Uhr beruht, in Frage gestellt. Insbesondere wurde der Wunsch nach einem geeigneten Zeitmodell zur Spezifikation von Echtzeitanforderungen formuliert. Aus Sicht der anderen wissenschaftlichen Partner (UPB, OFFIS) ist zudem eine Abbildung auf das kontinuierliche Zeitmodell, das den dort verwendeten Timed Automata zugrunde liegt, wünschenswert.
- *Modellierung dynamischer Systemstrukturen.* Von mehreren Partnern wurde der Wunsch nach geeigneten Konzepten zum Umgang mit dynamischen Systemstrukturen gewünscht. Während für die Automotive-Domäne nur die Betrachtung von zur Entwicklungszeit vordefinierten Rekonfigurationen entscheidend ist, ist für die Energie-Domäne auch der Umgang mit dynamischen Systemstrukturen zur Laufzeit relevant.
- *Integration verschiedener Ingenieursdisziplinen.* Da eingebettete Systeme mehr als nur Software umfassen und Hardware sowie regelungstechnische, elektronische und mechanische Aspekte enthalten, müssen Schnittstellen zu den jeweiligen Ingenieursdisziplinen geschaffen werden, um sie mit der softwaretechnischen Entwicklung zu verbinden.
- *Modellierung stochastischer Eigenschaften.* Aus Sicht der Energie-Domäne ist ferner die Modellierung stochastischer Eigenschaften bzw. Unschärfe von Interesse.
- *Modellierung nicht-funktionaler Eigenschaften.* Des weiteren kam der Wunsch nach einer geeigneten Modellierung von nicht-funktionalen Anforderungen auf.

Im Folgenden werden die Kommentare der einzelnen Feedback-Partner gegliedert nach Anwendungsdomänen zusammengefasst. Die Originalkommentare der jeweiligen Partner - soweit schriftlich vorhanden - finden sich im Anhang.

2 Automatisierungstechnik

Im Folgenden sind die wichtigsten Punkte des Feedbacks der Automatisierungsdomäne dargestellt, basierend auf dem Dokument [Buc09] von Christian Buckl (FORTISS/TUM-RES) (siehe Anhang A.3).

- *Zeitliches Verhalten.* Die vorgeschlagene Modellierungstheorie basiert auf einem zeitdiskreten Ausführungsmodell mit globaler Zeit, das synchronen Systemen sehr ähnelt. Während diese Systeme erwiesenermaßen wesentliche Vorteile in Bezug auf den Einsatz formaler Methoden besitzen, werden dadurch deutliche Einschränkungen in Bezug auf die Implementierung in Kauf genommen. Insbesondere wird die *Granularität* der

Zeitauflösung maßgeblich durch den Synchronisationsfehler bestimmt. Dies kann zu einem Widerspruch mit den Zeitanforderungen an das System stehen.

- *Berücksichtigung von Ressourcen.* Eingebettete Systeme zeichnen sich durch die Ausführung auf ressourcenbeschränkten Komponenten aus. Insbesondere steht die effiziente Ausnutzung der zur Verfügung stehenden Ressourcen im Vordergrund. Dies steht im Gegensatz zur Modellierungstheorie, die sich eher an der funktionalen Programmierung zu orientieren zu scheint. So würde eine Implementierung des laufenden Beispiels in eingebetteten Systemen sicherlich nicht durch eine parallele Ausführung der AND und OR Komponenten mit nachfolgender Auswahl des Ergebnisses erfolgen, sondern direkt die relevante Komponente ausführen. Hierzu notwendige Konstrukte wie ein Alternativ-Operator fehlen in der Modellierungstheorie. Eine Kombination des Datenflusses mit Konzepten wie Zustandsautomaten mit Parallel- und Alternativstatements analog zur IEC 61131-3 würde hier Abhilfe schaffen.

  Des Weiteren ist eine Spezifikation der für eine Ausführung von Komponenten benötigten Ressourcen in eingebetteten Systemen dringend erforderlich. Dies umfasst neben dem Prozessor auch weitere Betriebsmittel (z.B. Speicher, I/O, Semaphor, etc.).

- *Modularität & Komposition.* Ein wesentliches Ziel der Modellierungstheorie ist die Unterstützung von Modularität und Komposition. In der derzeitigen Umsetzung ist dies jedoch nur für die funktionalen Eigenschaften nicht jedoch für nicht-funktionale Eigenschaften (z.B. QoS) und das zeitliche Verhalten möglich. In Bezug auf nicht-funktionale Eigenschaften ist eine solche Forderung sicherlich nur sehr komplex umzusetzen, insofern sollte hiervon zunächst Abstand genommen werden. Dies gilt jedoch nicht für das zeitliche Verhalten. Kritikpunkte sind hier einerseits die Notwendigkeit schon im laufenden Beispiel (offensichtlich) willkürliche Entscheidungen zum Zeitverhalten einzelner Dienste zu treffen, um auf das gewünschte Ergebnis zu kommen. Zudem ist natürlich ein gleiches Verhalten der Komposition aufgrund der Abstützung auf gleiche Ressourcen (CPU) gar nicht möglich. Hier müssen Abstraktionsebenen, wie z.B. das Konzept der Fixed Logical Execution Times, eingeführt werden, um Komposition in Maßen zu unterstützen.

- *Nicht-funktionale Eigenschaften* Im Zusammenhang der Modellierungstheorie werden nicht-funktionale Eigenschaften nicht thematisiert.

3 Automotive

Der folgende Abschnitt umfasst die wichtigsten Feedback-Punkte der Automtovie-Domäne, basierend auf den Einträgen im SPES-Wiki [spe] und dem Dokument [Hol09] von Jörg Holtmann (UPB/Hella) (siehe Anhang A.1 und A.2).

- *Echtzeitanforderungen.* Wie kann man in FOCUS Echtzeitbeschränkungen formulieren? Was bieten mir die Nachrichtenströme diesbezüglich für Analysemöglichkeiten?

- *Kontinuierliches / hybrides Verhalten.* Aus dem Automotive-Bereich ist bekannt, dass Bosch diskrete und kontinuierliche Aspekte gleichermaßen betrachten will. Für bestimmte Entwicklungszweige von Hella ist dies ebenfalls interessant, allerdings nur wenn man mit solchen Modellen auch mehr erreichen kann als nur zu modellieren, z.B. Analysen fahren oder Code synthetisieren.

- *Kontinuierliches Zeitmodell.* Kontinuierliche Zeit ist nach Clarke das natürliche Modell für asynchrone Prozesse, welche typischerweise bei verteilten Systemen auftreten (da z.B. a-priori-Diskretisierung zu ungenauen Modellen führen kann und es bei kleinen Taktzyklen eher zur Zustandsraumexplosion bei Verifikation mit Model Checking kommt). Dieser Punkt ist ebenfalls wichtig für die Modellierung kontinuierlicher/hybrider Systeme sowie die Integration mit den Modellierungstheorien Rich Components und Mechatronic UML.
- *Lokale Uhren.* Wie passt eine globale Uhr zusammen mit der Annahme von verteilten Systemen?
- *Modemanagement.* Rekonfigurationen die zum Entwicklungszeitpunkt bekannt sind.

Von Hans-Werner Wiesbrock (IT Power Consultants)

- In [HHR09, Seite 11] steht zu dem Communication Paradigm: ... the sender can write a message without delay... ...usually connected to a bus which never blocks the sender. Im Automotive Bereich ist der CAN Bus sehr prominent und die meisten Steuergeräte im Fahrzeug kommunizieren darüber. Das CAN Protokoll arbeitet u.a. mit Priorisierungen von Botschaften, der Zeitpunkt ihres Versendens hängt davon ab. Mehr noch, es kann auch zu den bekannten Priorisierungsinversionen kommen. Insbesondere im letzteren Fall zeigt sich, dass eine Botschaft hier sehr wohl geblockt werden kann, leider theoretisch beliebig lange.

4 Energie

Im folgenden Abschnitt werden die wichtigsten Feedback-Punkte der Energie-Domäne dargestellt, basierend auf dem Dokument von Martin Fritzsche (TUM-SSE/SWM) [Fri09] (siehe Anhang A.1).

- *Dynamische Systemstruktur.* Die Möglichkeit dynamischer Veränderungen in der Systemstruktur sollte im Modell darstellbar sein. Zum Beispiel können zu einem Erzeugungsanlagenpark jederzeit Kleinstkraftwerke hinzukommen oder wegfallen.
- *Stochastische Prozesse.* Es sollte möglich sein, in dem Modell stochastische Prozesse darzustellen, sprich Ereignisse, die mit einer bestimmten Wahrscheinlichkeit auftreten. Zum Beispiel können Komponenten und Kommunikationswege ausfallen. Bei der Konzeption von Steuerkomponenten spielt es eine Rolle, diese Ausfallwahrscheinlichkeiten zu kennen und mit zu berücksichtigen. Oder folgen Komponenten nur mit einer bestimmten Wahrscheinlichkeit dem vorgegebenen Verhalten.
- *Lokale Uhren.* Die Theorie geht von der Existenz einer globalen Uhr aus. Das ist eine Annahme, die in einem massiv verteilten System wie den in der Energiedomäne betrachteten Smart Grids nicht haltbar ist.
- *Integration von verschiedenen Ingenieursdisziplinen* Da eingebettete Systeme mehr als nur Software umfassen und Hardware sowie regelungstechnische, elektronische und mechanische Aspekte enthalten können, müssen Schnittstellen zu den jeweiligen Ingenieursdisziplinen geschaffen werden, um sie mit der softwaretechnischen Entwicklung zu verbinden.

5 OFFIS

Im folgenden Abschnitt wird der wichtigste Feedback-Punkt von OFFIS dargestellt, basierend auf diversen Diskussionsrunden zu dem Thema Modellierungstheorie.

- *Echtzeitverhalten.* Der Rich-Components-Ansatz basiert auf Timed Automaten und demzufolge unterstützt das Echtzeitverhalten.

A Original Feedback und Änderungswünsche

A.1 Sammlung von Feedback und Änderungswünsche im SPES-Wiki, Stand 13.11.2009

Seit 21.08.2009 stand im SPES-Wikipedia unter https://spes.informatik.tu-muenchen.de/index.php/ZP-AP_1.1._Liste_von_%C3%84nderungsw%C3%BCnschen_an_die_Modellierungstheorie eine Seite zur Sammlung von weiteren Anforderungen und Änderungswünschen an die Modellierungstheorie bereit. Die nachfolgende Tabelle 1 zeigt den Stand am 13.11.2009 auf:

Name/Organisation	Datum	Änderungswunsch
Dr. Hans-Werner Wiesbrock / IT Power Consultants	27.08.2009	Auf Seit 11 schreiben Sie zu dem Communication Paradigm: ... the sender can write a message without delay... ...usually connected to a bus which never blocks the sender. Im Automotive Bereich ist der CAN Bus sehr prominent und die meisten Steuergeräte im Fahrzeug kommunizieren darüber. Das CAN Protokoll arbeitet u.a. mit Priorisierungen von Botschaften, der Zeitpunkt ihres Versendens hängt davon ab. Mehr noch, es kann auch zu den bekannten Priorisierungsinversionen kommen. Insbesondere im letzteren Fall zeigt sich, dass eine Botschaft hier sehr wohl geblockt werden kann, leider theoretisch beliebig lange. Wie verträgt sich das CAN Protokoll mit Ihrer Annahme?

Jörg Holtmann / UPB,Hella	27.10.2009	Änderungswünsche: ■ Real-Time Constraints (S.6): Wie kann man in FOCUS Echtzeitbeschränkungen formulieren? Was bieten mir die Nachrichtenströme diesbezüglich für Analysemöglichkeiten? ■ Continuous / hybrid Modeling (S.6): Aus dem Automotive-Bereich ist bekannt, dass Bosch diskrete und kontinuierliche Aspekte gleichermaßen betrachten will. Für bestimmte Entwicklungszweige von Hella ist dies ebenfalls interessant, allerdings nur wenn man mit solchen Modellen auch mehr erreichen kann als nur zu modellieren, z.B. Analysen fahren oder Code synthetisieren. ■ Dynamic Reconfiguration (S.6): Soweit ich weiß soll "'Modemanagement"' unterstützt werden, d.h. Rekonfigurationen die zum Entwicklungszeitpunkt bekannt sind. Daher sollte man unter den adressierten Aspekten einen Punkt wie "'Statische Rekonfiguration"' oder "'Modemanagement"' hinzufügen und die Punkte voneinander abgrenzen. ■ Timing Model (S.12): Kontinuierliche Zeit ist nach Clarke das natürliche Modell für asynchrone Prozesse, welche typischerweise bei verteilten Systemen auftreten (da z.B. a-priori-Diskretisierung zu ungenauen Modellen führen kann und es bei kleinen Taktzyklen eher zur Zustandsraumexplosion bei Verifikation mit Model Checking kommt). Dieser Punkt ist ebenfalls wichtig für die Modellierung kontinuierlicher/hybrider Systeme sowie die Integration mit den Modellierungstheorien Rich Components und Mechatronic UML. ■ Synchronität (S.12): Wie passt eine globale Uhr zusammen mit der Annahme von verteilten Systemen?

Martin Fritzsche / SWM	05.11.2009	Änderungswünsche: ■ Die Möglichkeit dynamischer Veränderungen in der Systemstruktur sollte im Modell darstellbar sein (z.B. zu einem Erzeugungsanlagenpark können jederzeit Kleinstkraftwerke hinzukommen oder wegfallen). ■ Es sollte möglich sein in dem Modell stochastische Prozesse darzustellen, sprich Ereignisse, die mit einer bestimmten Wahrscheinlichkeit auftreten. ■ Die FOCUS-Theorie geht von der Existenz einer globalen Uhr aus. Das ist eine Annahme, die in einem massiv verteilten System wie den in der Energiedomäne betrachteten Smart Grids nicht haltbar ist.

Table 1: Sammlung von Änderungswünschen an die Modellierungstheorie im SPES-Wiki, Stand 13.11.2009

A.2 Anforderungen an Modellierungstheorie von Jörg Holtmann (UPB/Hella)

Das Deliverable AU-D.3.1.A-1 [Hol09] greift die initialen Anforderungen an die zu entwickelnde, gemeinsame Modellierungstheorie aus dem Dokument [HHR09] auf und erweitert sie aus Sicht der Automotive-Domäne.

Software Plattform Embedded Systems 2020

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

- Anforderungen an eine durchgängige Modellierungstheorie für eingebettete Systeme im Automobilsektor -

Deliverable AU-D.3.1.A-1

Version: 1.0

Projektbezeichnung	SPES 2020	
Verantwortlich	Jörg Holtmann	
QS-Verantwortlich	Hella KGaA Hueck & Co.	
Erstellt am	30.07.2009	
Zuletzt geändert	04.11.2009 18:06	
Freigabestatus		Vertraulich für Partner: s-lab/Universität Paderborn; Hella KGaA Hueck & Co.
	X	Projektöffentlich
		Öffentlich
Bearbeitungszustand		in Bearbeitung
		vorgelegt
	X	fertig gestellt

Weitere Produktinformationen

Erzeugung	Jörg Holtmann
Mitwirkend	Jan Meyer, Ulrich Nickel

Änderungsverzeichnis

Änderung			Geänderte Kapitel	Beschreibung der Änderung	Autor	Zustand
Nr.	Datum	Version				
1	20.07.09	0.1	Alle	Initiale Produkterstellung	JH	in Bearbeitung
1	30.07.09	0.2	Alle	Einarbeitung Kommentare internes Review	JH	vorgelegt
2	23.10.09	0.3	Alle	Neue Formatvorlage	JH	vorgelegt
3	30.10.09	1.0	3.9	Überarbeitung	UN	Fertig gestellt

Kurzfassung

Der Entwicklungsprozess für eingebettete Systeme unterliegt im Automobilsektor einigen speziellen Charakteristika, wie z.B. einer hohen Systemheterogenität, einem hohen Maß an Interaktion zwischen den einzelnen Teilsystemen, Sicherheitsanforderungen und einem hohen Kostendruck. Dieses Dokument greift die initialen Anforderungen an die zu entwickelnde, gemeinsame Modellierungstheorie für das SPES-Projekt aus dem SPES Deliverable D.1.1.A-1 [HRT09] auf und erweitert sie im Bezug auf die genannten Charakteristika.

Inhalt

1 Einordnung und Kurzbeschreibung

Dieses Kapitel fasst das vorliegende Dokument zusammen und ordnet es in den SPES Projektkontext ein. Anschließend wird ein Überblick über die Dokumentstruktur gegeben.

1.1 Motivation und Einordnung

Das vorliegende Dokument ist Teil des Deliverable AU-D.3.1.A für die Task 3.1 im Arbeitspaket 3 des Anwendungsprojekts Automotive von SPES 2020. In diesem Arbeitspaket wird, ähnlich wie im gesamten SPES-Projekt, eine durchgängige Entwicklungsmethodik für eingebettete Systeme erforscht, allerdings zugeschnitten auf die Bedürfnisse von Hella KGaA Hueck & Co. und damit auf die Zuliefererindustrie des Automobilsektors. Dieses Dokument stellt unter diesen Rahmenbedingungen die von Hella KGaA Hueck & Co. erhobene Anforderungen an die in SPES 2020 zu entwickelnde Modellierungstheorie vor. Dabei wird das SPES-Deliverable D.1.1.A-1 [HRT09] als Grundlage genommen und um eigene Aspekte erweitert. Eine erweiterte Version dieses Dokuments, für das Anforderungen von der Robert Bosch GmbH erhoben werden, ist geplant.

Das Teil-Deliverable AU-D.3.1.A-2 [Hol09b] wurde ebenfalls in Zusammenarbeit mit Hella KGaA Hueck & Co. erarbeitet und geht auf die im Entwicklungsprozess eingebetteter Systeme typischerweise verwendeten Abstraktionsebenen ein. Zusammen mit diesem Dokument vervollständigt es das Deliverable AU-D.3.1.A.

1.2 Management Summary

Der Entwicklungsprozess für eingebettete Systeme unterliegt im Automobilsektor einigen speziellen Charakteristika. Zum einen herrscht aufgrund der typischen Zulieferer-Hersteller-Beziehung und des aktuell noch unzulänglichen Architekturstandards AUTOSAR ein hohes Maß an Heterogenität der verschiedenen Systeme. Diese heterogenen Systeme müssen beim Automobilhersteller integriert werden. Zusätzlich werden diese einzelnen Systeme oft miteinander verschaltet, um weiterführende Funktionen zu erfüllen als es die Einzelsysteme könnten. Des Weiteren nehmen eingebettete Systeme in Fahrzeugen oft eine sicherheitskritische Rolle ein und müssen neben einer korrekten Funktionsweise harte Echtzeitanforderungen erfüllen. Letztendlich führt die Massenfertigung von Fahrzeugen zu einem hohen Kostendruck, da geringe Ersparnisse für ein einzelnes Steuergerät hohe Ersparnisse bzgl. der Kosten einer kompletten Fahrzeugserie bedeuten. Dieses Dokument greift die Anforderungen an die zu entwickelnde, gemeinsame Modellierungstheorie für das SPES-Projekt aus dem SPES-Deliverable D.1.1.A-1 [HRT09] auf und erweitert sie unter Bezug auf die zuvor genannten Charakteristika.

1.3 Überblick

Das erste Kapitel grenzt die Charakteristiken der zu betrachtenden Systeme ein, während das zweite Kapitel die eigentlichen Anforderungen an die Modellierungstheorie anführt.

2 Charakteristiken adressierter Systeme

Dieser Abschnitt beschreibt, welche Aspekte der zu modellierenden Systeme von der zukünftigen Modellierungstheorie adressiert werden sollen und welche Aspekte in der Automobilindustrie aktuell sowie in näherer Zukunft nicht benötigt werden.

2.1 Welche Aspekte sollen adressiert werden?

In diesem Abschnitt werden die Aspekte der Systeme genannt, welche von der Modellierungstheorie unterstützt werden sollen.

2.1.1 Multifunktionale und komplexe Systeme

Systeme, welche heutzutage im Automobilsektor in der Zuliefererindustrie entwickelt werden, besitzen von sich aus bereits diverse Funktionalitäten. Somit müssen also zum einen *multifunktionale Systeme* betrachtet werden.

Zum anderen werden manche dieser Systeme bereits beim Zulieferer miteinander kombiniert, um erweiterte Funktionalitäten zu ermöglichen. Hinzu kommt, dass diese Systeme von den Automobilherstellern integriert werden. Teilweise werden sie auch miteinander zu komplexeren Systemen verschaltet, so kann ein Tempomat als Einzelsystem fungieren, aber auch in Kombination mit einem Abstandssensor für eine adaptive Geschwindigkeitsregelung eingesetzt werden. Durch diese Interaktionen der Funktionalitäten ergeben sich also zusätzlich *komplexe Systeme*.

2.1.2 Interaktive und reaktive Systeme

In Automobilen gibt es beide Arten von eingebetteten Systemen, *interaktive Systeme* zur Interaktion mit dem Benutzer und *reaktive Systeme* zur Reaktion auf Umgebungseinflüsse. Somit müssen diese beiden Arten von Systemen betrachtet werden.

2.1.3 Verteilte Systeme

In aktuellen Fahrzeugen existieren diverse Systeme mit dedizierten Funktionalitäten, welche auf Steuergeräten über das gesamte Fahrzeug verteilt sind. Auf den Steuergeräten laufen eigenständige Anwendungen, welche über keinen gemeinsamen Speicher verfügen. Zwischen den einzelnen Anwendungen, deren Steuergeräte über Busse miteinander verbunden sind, wird i.A. über einen asynchronen Nachrichtenversand kommuniziert, wie z.B. beim CAN-Bus. Daher müssen also *verteilte Systeme* berücksichtigt werden.

2.1.4 Diskrete Steuergeräte, quasikontinuierliche Regler und kontinuierliche physische Prozesse

Innerhalb der Softwaretechnik erstellte Steuergeräte zur Beschreibung von reaktivem Verhalten stellen i.A. eine Form von Zustandsmaschinen dar, was in *zeitdiskretem und zustandsabhängigem Verhalten* resultiert. Auf diesen Systemen liegt der Hauptfokus von SPES.

Des Weiteren trifft man im Bereich der eingebetteten Systeme auf in der Regelungstechnik entwickelte Regler, deren Verhalten mit kontinuierlichen Modellen wie z.B. Differentialgleichungen spezifiziert wird. Man unterscheidet die Analog- und Digitalregler. Erstere könne Signale mit unendlicher Auflösung verarbeiten und sind technisch nicht im Kontext von SPES zu berücksichtigen. Für Digitalregler hingegen wird die kontinuierliche Verhaltensbeschreibung diskretisiert. Da hier eine Approximation

des kontinuierlichen Verhaltens stattfindet und die Abtastrate für solche Systeme weitaus geringer ist als die für zeitdiskrete und zustandsabhängige Systeme ist, spricht man hier auch von *quasi-kontinuierlichem Verhalten*. Solche Regler sollten ebenfalls in gewisser Form berücksichtigt werden, z.B. in Form von Schnittstellen zu den kontinuierlichen Modellen.

Diese verschiedenen Systemarten sind wiederum mit *kontinuierlichen, physischen Prozessen* aus der Umwelt konfrontiert. Dies ist bei der Entwicklung von eingebetteten Systemen zu beachten.

2.1.5 Systeme mit Echtzeitbeschränkungen

Viele Funktionen in Automobilen unterliegen Echtzeitanforderungen. Dies bedeutet, dass sie nicht nur korrekt funktionieren sollen, sondern auch innerhalb einer vorhersagbaren Zeit.

Dabei ist zwischen Systemen mit *weichen* und *harten Echtzeitanforderungen* zu unterscheiden. Weiche Echtzeitanforderungen liegen z.B. bei Multimediawiedergabegeräten vor, die zwar Bild und Ton innerhalb gewisser Zeitschranken wiedergeben müssen. Ein kurzzeitiges Überschreiten dieser Schranken führt zu kleinen, mehr oder weniger wahrnehmbaren Rucklern in der Wiedergabe. Eine solche kurzfristige Überschreitung wird nicht als kritisch bzw. als Fehler angesehen. An sicherheitskritische Systeme werden typischerweise harte Echtzeitanforderungen gestellt, d.h. dass ein einmaliges und kurzfristiges Überschreiten einer Zeitschranke zu einem Unfall führen kann, bei dem Mensch und Umwelt in Gefahr geraten. Beispiele hierfür sind das rechtzeitige Einsetzen eines Antiblockiersystems oder die rechtzeitige Auslösung von Airbags.

Da es für eingebettete Systeme in Automobilen beide Arten von Echtzeitanforderungen gibt, sind also *Systeme mit Echtzeitbeschränkungen* zu berücksichtigen. Da die harten Echtzeitanforderungen starken Einfluss auf die Sicherheit der Fahrzeuginsassen und der Umgebung haben, sind diese zu fokussieren.

2.1.6 Hybride Systeme

Im Automobilsektor lassen sich vor allem im Bereich von Motorsteuerungen und Bremssystemen kontinuierliche Verhaltensaspekte nicht ausblenden. Um Konzepte von unterschiedlichen „Disziplinen“ wie Softwaretechnik und Regelungstechnik zu vereinen, welche im Bereich der eingebetteten Systeme oft zusammen auftreten, ist es daher erstrebenswert, sowohl diskretes als auch kontinuierliches Verhalten modellieren zu können oder zumindest eine Schnittstelle zwischen der diskreten und der kontinuierlichen Welt zur Verfügung zu stellen, um insgesamt ein hybrides Verhalten zu unterstützen [GH06] . Somit muss also ein Konzept zur Modellierung *hybrider Systeme* entwickelt werden.

2.1.7 Systeme mit vordefinierter Rekonfiguration

Systeme, welche Reglerumschaltungen bzw. Mode Changes und somit das Umschalten von vordefinierten Konfigurationen erlauben, werden bereits produktiv eingesetzt. Wenn man solche Rekonfigurationen soweit in der Modellierungstheorie verankern kann, dass sie nicht nur modellierbar, sondern auch simulierbar, analysierbar und/oder verifizierbar sind, dann ist es ebenso wünschenswert solche Systeme zu unterstützen. Dieser Punkt steht stark im Zusammenhang mit der Modellierung von hybriden Systemen.

2.2 Welche Aspekte müssen nicht adressiert werden?

Dieser Abschnitt führt Aspekte auf, welche in näherer Zukunft bei der Entwicklung von eingebetteten Systemen in der Automobilindustrie nicht berücksichtigt werden müssen.

2.2.1 Probabilistische Systeme

Im Allgemeinen möchte man ein deterministisches System spezifizieren, probabilistische Systeme sind eher ein Forschungskonstrukt. Ausnahmen finden sich in der quantitativen Gefahrenanalyse, in der man ausgehend von Fehlerauftrittswahrscheinlichkeiten die Auftrittswahrscheinlichkeit von aus den Fehlern resultierenden Gefahren im System bestimmen will. Das Verhalten des Systems wird dabei dennoch deterministisch spezifiziert.

2.2.2 Dynamisch rekonfigurierbare bzw. dynamische Systeme

Systeme, in denen zur Laufzeit Rekonfigurationen vorgenommen werden, welche zur Entwurfszeit nicht bekannt sind, werden im Automobilsektor aktuell nicht entwickelt. Die Entwicklung solcher Systeme erfordert andere Entwurfs- und Verifikationstechniken als die Entwicklung von Systemen mit vordefinierten Rekonfigurationen, wie sie in Abschnitt 2.1.7 beschrieben wurden. Abgesehen davon, dass die Spezifikation dieser dynamischen Systeme noch ein forschungsnahes Thema ist, gibt es verschiedene Standards und Normen (MISRA, IEC 61508, ISO 26262, …), welche den Einsatz von Modellierungstechniken für solche Zwecke einschränken bzw. in Frage stellen.

3 Anforderungen an die Modellierungstheorie

Im Folgenden werden, teilweise basierend auf den bereits beschriebenen zu betrachtenden Systemen, direkte Anforderungen an die Modellierungstheorie gestellt.

3.1 Modularität und Kompositionalität

Eingebettete Systeme im Automobilsektor können heutzutage aus Komplexitätsgründen nicht mehr als Ganzes entwickelt und analysiert werden, sondern man betrachtet i.A. einzelne Subsysteme mit dedizierten Funktionen und komponiert diese zu einem Gesamtsystem, welches wiederum Teil eines weiteren Systems sein kann. Des Weiteren lassen sich einmal entwickelte Subsysteme oder Komponenten in verschiedenen Systemen einsetzen, somit erhöht sich durch ein Modularitätskonzept die Wiederverwendbarkeit der Funktionalitäten solcher Komponenten. Für diesen Zweck muss vor allem auf eine wohldefinierte Schnittstellenabstraktion der Komponenten geachtet werden.

3.2 Abstraktion und Verfeinerung

Um den Zusammenhang und Übergang zwischen generischeren und konkreteren Modellebenen zu definieren, kann das Konzept der Verfeinerung herangezogen werden. Ein konkreteres Modellelement verfeinert dabei die Informationen eines abstrakteren, indem Informationen derart hinzugefügt werden, dass die Informationen des abstrakteren Modellelements erhalten bleiben.

Die Schnittstellenabstraktion soll es ermöglichen, dass Systeme unabhängig von ihrer Implementierung in verschiedenen Kontexten benutzt werden können. Dieses Prinzip ist vor allem Grundlage für die im letzten Unterabschnitt geforderte Modularität.

3.3 Explizite Modellierung von Zeit

Im Bereich der eingebetteten Systeme muss eine Modellierungstheorie die explizite Modellierung von Zeitaspekten unterstützen, um Systemen mit Echtzeitaspekten gerecht zu werden (siehe Abschnitt 2.1.5). In diesem Zusammenhang ist es vor allem wichtig, diese Zeitaspekte in einer möglichst frühen Designphase zu berücksichtigen, damit Entwurfsexplorationen und weitere Designentscheidungen ebenfalls möglichst früh und fundiert stattfinden können.

3.4 Unterstützung verschiedener „Sichten“

Ebenfalls aus Komplexitätsgründen ist die Unterstützung von verschiedenen „Sichten“ – wie Struktur, Verhalten, Deployment, Daten – empfehlenswert, damit die entsprechenden Aspekte getrennt voneinander betrachtet werden können.

3.5 Integration verschiedener Ingenieursdisziplinen

Da eingebettete Systeme mehr als nur Software umfassen und Hardware sowie regelungstechnische, elektronische und mechanische Aspekte enthalten können, müssen Schnittstellen zu den jeweiligen Ingenieursdisziplinen geschaffen werden, um sie mit der softwaretechnischen Entwicklung zu verbinden. Hierzu gehört auch ein passendes Abstraktionsebenenkonzept, welches in einer frühen Designphase das Gesamtsystem betrachtet und dieses in späteren Entwicklungsphasen innerhalb der einzelnen Disziplinen detailliert.

3.6 Durchgängigkeit und Nachvollziehbarkeit

Die Theorie sollte eine durchgängige Verknüpfung zwischen verschiedenen Modellelementen wie Anforderungen, strukturellen und verhaltensorientierten Artefakten, Testfällen etc. unterstützen, um Nachvollziehbarkeit zu gewährleisten – insbesondere auch modellübergreifend.

3.7 Anpassbarkeit

Der Ansatz sollte domänenspezifisch instanziierbar sein, d.h. es sollten verschiedene Spezifikationsbeschreibungen einsetzbar und das Metamodell erweiterbar sein.

3.8 Verifizierbarkeit

Die Theorie muss verifizierbare und analysierbare Modelle hervorbringen, eine Spezifikation von Modellen zur bloßen Dokumentation reicht nicht aus.

3.9 Kontinuierliches Zeitmodell

Da verteilte Systeme betrachtet werden, hat man es mit asynchron agierenden Prozessen zu tun. Für solche Systeme ist kontinuierliche Zeit das natürliche Modell. Soll die Modellierungstheorie Basis für die Analyse, Simulation oder formale Verifikation der betrachteten Systeme dienen, dann kann die Wahl eines diskreten Zeitmodells für asynchrone Systeme zu diversen Problemen führen [CGP99] . So erfordert zum einen die Verwendung diskreter Zeitmodelle, dass Taktzyklen für die zu modellierenden Systeme bereits a priori festgelegt werden. Dies beschränkt die Genauigkeit der resultierenden Modelle. In der Praxis können hier ggf. relevante Effekte durch ein zu grobes Zeitraster im Streckenmodell übersehen werden. Will man diese Effekte (z.B. Fehlerfälle, welche durch mechanische Effekte wie die Verklemmung eines Aktuators hervorgerufen werden) in einem zeitdiskreten Streckenmodell untersuchen, dann bedingt dies teilweise, dass die Zeitbasis eine sehr hohe Auflösung haben muss. Dies ist aber wiederum für die Untersuchung des Normalbetriebs häufig nicht notwendig. Die Auswahl eines ausreichend kleinen Taktzyklus für die Modellierung eines asynchronen Systems kann aber zu sehr langen Rechenzeiten bei der Simulation oder zu einer Zustandsraumexplosion bei einer Verifikation über Model Checking führen.

4 Zusammenfassung

In diesem Dokument wurden als Teil des Deliverable AU-D.3.1.A im Bezug auf [HRT09] Anforderungen der Zulieferer im Automobilsektor an die in SPES 2020 zu entwickelnde Modellierungstheorie erhoben. Die Anforderungen berücksichtigen dabei die speziellen Charakteristika des Entwicklungsprozesses von eingebetteten Systemen im Automobilsektor.

Im ersten Kapitel wurden die zu betrachtenden Systeme durch eine Auflistung aller relevanten und irrelevanten Aspekte eingegrenzt. Im zweiten Kapitel wurden aus den relevanten Aspekten die konkreten Anforderungen an die Modellierungstheorie abgeleitet.

Insgesamt ist zu festzustellen, dass die Anforderungen an die Modellierungstheorie aus dem Deliverable D.1.1.A-1 [HRT09] bereits in großen Teilen mit den in diesem Dokument vorgestellten Anforderungen übereinstimmen. Verbesserungs- bzw. Erweiterungsbedarf seitens des Anwendungsprojekts Automotive besteht in den Punkten Integration verschiedener Ingenieursdisziplinen, Duchgängigkeit/Nachvollziehbarkeit, Anpassbarkeit, Verifizierbarkeit und kontinuierliches Zeitmodell. Des Weiteren sollten zusätzlich hybride Systeme betrachtet werden.

5 Literaturverzeichnis

[CGP99] Clarke Jr., E. M., Grumberg, O., & Peled, D. A. (1999). *Model Checking.* The MIT Press.

[GH06] Giese, H., & Henkler, S. (December 2006). A Survey of Approaches for the Visual Model-Driven Development of Next Generation Software-Intensive Systems. *Journal of Visual Languages and Computing , 17* (6), S. 528-550.

[HRT09] Harhurin, A., Hartmann, J., & Ratiu, D. (2009). *Modeling Theory - Motivation and Introduction of a Comprehensive Modeling Theory for Embedded Systems.* SPES 2020 Deliverable D.1.1.A-1, Version 1.2.

[Hol09b] Holtmann, J. (2009). *Anforderungen an Abstraktionsebenen im Entwicklungsprozess eingebetteter Systeme im Automobilsektor.* SPES 2020 Deliverable AU-D.3.1.A-2, Version 0.3, s-lab/Universität Paderborn.

A.3 Feedback und Änderungswünsche von Christian Buckl (FORTISS/TUM-RES)

Das Dokument [Buc09] umfasst das Feedback und die Änderungswünsche aus Sicht der Automatisierungsdomäne.

Dieses Dokument kommentiert die Modellierungstheorie, wie vom Lehrstuhl Prof. Broy im Rahmen des Deliverables D1.1.A-1 (Version 1.1 vom 21.08.2009) vorgeschlagen, aus Sicht der Partner der Anwendungsdomäne Automatisierungstechnik. Die Anmerkungen beziehen sich dabei nicht rein auf das Anwendungsgebiet Automatisierungs- und Produktionstechnik, sondern haben eine allgemeine Bedeutung für den Bereich eingebetteter Systeme. Generell wird vorgeschlagen das Dokument „Theory for Embedded Software“ und nicht „Theory for Embedded Systems“ zu nennen, da der Fokus ausschließlich auf der Software liegt. Des Weiteren sollte betont werden, dass das Dokument nur die zugrundliegende Modellierungstheorie, nicht jedoch die komplette Modellierungstheorie adressiert, die nach Meinung der Autoren dieses Kommentars auch alle Modelle bis zu den domänenspezifischen Modellen und deren Abbildung enthalten müsste.
Insgesamt ist festzustellen, dass der vorstellten Modellierungstheorie eine anderes Grundverständnis von Eingebetteten Systemen (im Sinne (Embedded) Software) zu Grunde liegt, als vielen der Kommentare (Embedded System im Sinne Informationsverarbeitung + Material/Energytransformation + Schnittstellen dazwischen). Diesbezüglich ist dringend weitere Klärung / Abgrenzung erforderlich.

Struktur:
Abschnitt 1 definiert die wesentlichen Fragestellungen, die aus derzeitiger Sicht in der Modellierungstheorie nicht hinreichend berücksichtigt wurden. Abschnitt 2 kommentiert im Detail einzelne Abschnitte des vom Lehrstuhl Prof. Broy vorgestellten Dokumentes. Abschnitt 3 enthält schließlich eine Auflistung von Korrekturvorschlagen in Bezug auf Inkonsistenzen und Rechtschreibfehler.

1. Unberücksichtigte / mangelhafte Aspekte eingebetteter Systeme

Ein wesentlicher Kritikpunkt an der Modellierungstheorie ist die Inkonsistenz der Abschnitte 2 und 3 mit der vorgeschlagenen Modellierungstheorie. Viele Aspekte werden korrekterweise als Anforderung identifiziert, jedoch nicht in der Modellierungstheorie berücksichtigt. Insbesondere die für eingebettete Systeme relevanten Themenbereiche zeitl. Verhalten, eingeschränkte Ressourcen, Hardware, nicht-funktionale Anforderungen, sowie die Thema der Unterstützung verschiedener Sichten und Modularisierung/Komposition für einen effizienten Entwicklungsprozess werden in Kapitel nur unzureichend / nicht adressiert.
Im Folgenden werden diese einzelnen Punkte näher erläutert und Anforderungen, sowie teilweise auch Lösungen vorgeschlagen.

Zeitliches Verhalten:
Die vorgeschlagene Modellierungstheorie basiert auf einem zeitdiskreten Ausführungsmodell mit globaler Zeit (siehe S. 12), das synchronen Systemen sehr ähnelt. Während diese Systeme erwiesenermaßen wesentliche Vorteile in Bezug auf den Einsatz formaler Methoden besitzen, werden dadurch deutliche Einschränkungen in Bezug auf die Implementierung in Kauf genommen. Insbesondere wird die **Granularität** der Zeitauflösung maßgeblich durch den Synchronisationsfehler bestimmt. Dies kann zu einem Widerspruch mit den Zeitanforderungen an das System stehen.
Noch schwerwiegender ist dagegen der Ausschluss von **Determinismus** in Bezug auf das Zeitverhalten. So kann zwar das funktionelle Verhalten durch Unterspezifikation nicht deterministisch (oder zumindest nicht direkt) modelliert werden, dies trifft jedoch nicht auf das zeitliche Verhalten zu. Durch die Modellierungstheorie wird das Zeitverhalten durch die Latenz in Bezug auf Zeitticks spezifiziert. Dabei ist die Anzahl an Ticks exakt angegeben und die einzige Möglichkeit Varianzen in Bezug auf das zeitliche Verhalten anzugeben ist die explizite Modellierung aller möglichen Verhalten. Dies führt zu einer aus unserer Sicht nicht

handhabbaren Explosion der zu modellierenden Verhalten. Insbesondere in der Requirementsphase, jedoch auch zur Laufzeit muss jedoch auch Nicht-Determinismus in Bezug auf das zeitliche Verhalten möglich sein. Ein einfaches und offensichtliches Beispiel ist das Verhalten der Kommunikation mit minimalen und maximalen (im Fall von echtzeitfähigen Kommunikationsprotokollen) Latenzzeiten. Eine Möglichkeit zur Spezifikation von Nicht-Determinismus wäre die Abstützung auf Konzepten aus der temporalen Logik. Des Weiteren hat sich eine Unterscheidung zwischen physikalischer und logischer Zeit (siehe z.B. Projekt Ptides) als hilfreich zur Modellierung des Zeitverhaltens erwiesen. Dabei dient die logische Zeit ausschließlich zur Beschreibung von zeitlichen Zusammenhängen(zeitliche Anordnung) der Komponente, das heißt sie beinhaltet nur Teilaspekte des Zeitverhaltens, und zwar die Kausalität.
Auch die Definition 5, siehe späterer Kommentar, widerspricht dem Charakter von Echtzeitsystemen die durch eine ständige und insbesondere **unwiderruflichen Interaktion mit der Umwelt** gekennzeichnet sind und zeugt von der Abstützung der Theorie auf den Bereich der funktionalen Programmierung.

Berücksichtigung von Ressourcen:
Eingebettete Systeme zeichnen sich durch die Ausführung auf ressourcenbeschränkten Komponenten aus. Insbesondere steht die **effiziente Ausnutzung** der zur Verfügung stehenden Ressourcen im Vordergrund. Dies steht im Gegensatz zur Modellierungstheorie, die sich eher an der funktionalen Programmierung zu orientieren zu scheint. So würde eine Implementierung des laufenden Beispiels in eingebetteten Systemen sicherlich nicht durch eine parallele Ausführung der AND und OR Komponenten mit nachfolgender Auswahl des Ergebnisses erfolgen, sondern direkt die relevante Komponente ausführen. Hierzu notwendige Konstrukte wie ein Alternativ-Operator fehlen in der Modellierungstheorie. Eine Kombination des Datenflusses mit Konzepten wie Zustandsautomaten mit Parallel- und Alternativstatements analog zur IEC 61131-3 würde hier Abhilfe schaffen.
Des Weiteren ist eine Spezifikation der für eine Ausführung von Komponenten benötigten **Ressourcen** in eingebetteten Systemen dringend erforderlich. Dies umfasst neben dem Prozessor auch weitere Betriebsmittel (z.B. Speicher, I/O, Semaphor, Energie, etc.).

Modularität & Komposition
Ein wesentliches Ziel der Modellierungstheorie ist die Unterstützung von Modularität and Komposition (siehe S. 7). In der derzeitigen Umsetzung ist dies jedoch nur für die funktionalen Eigenschaften nicht jedoch für nicht-funktionale Eigenschaften (z.B. QoS) und das zeitliche Verhalten möglich. In Bezug auf nicht-funktionale Eigenschaften ist eine solche Forderung sicherlich nur sehr komplex umzusetzen, insofern sollte hiervon zunächst Abstand genommen werden. Dies gilt jedoch nicht für das zeitliche Verhalten. Kritikpunkte sind hier einerseits die Notwendigkeit schon im laufenden Beispiel (offensichtlich) willkürliche Entscheidungen zum Zeitverhalten einzelner Dienste zu treffen, um auf das gewünschte Ergebnis zu kommen (siehe S. 20). Zudem ist natürlich ein gleiches Verhalten der Komposition aufgrund der Abstützung auf gleiche Ressourcen (CPU) gar nicht möglich. Hier müssen Abstraktionsebenen, wie z.B. das Konzept der Fixed Logical Execution Times, eingeführt werden, um Komposition in Maßen zu unterstützen.

Nicht-funktionale Eigenschaften und Sichten
Im Zusammenhang der Modellierungstheorie werden nicht-funktionale Eigenschaften, sowie Sichten, letzteres trotz Erwähnung im Kapitel Anforderungen (S.8) nicht thematisiert. Dies umfasst insbesondere die Einwirkung der Umwelt auf die Software (z.B. Fehler), aber auch die Auswirkungen aus der Interaktion mit der Hardware (siehe Berücksichtigung von Ressourcen).

Hardware
Die Hardware eines Eingebetteten Systems stellt einen ebenso bedeutenden Systemaspekt dar, wie dessen Software. Eine umfassende Modellierungstheorie (für die Software) muss diesem Aspekt Rechnung tragen, indem damit auch zumindest für die Software relevanten Hardwarekomponenten (Rechner, Kommunikationskanäle) und ihre wesentlichen Eigenschaften modelliert werden können.

Modellierung der (Beziehungen zur) Umgebung
In Kapitel 3 wird folgende Anforderung genannt: „Embedded systems are frequently used to control processes and devices that consist of physical (e. g., mechanical, electrical) components and exhibit time-continuous behavior." Allerdings bietet die Modellierungstheorie keine Konzepte an, diese Komponenten als Umgebung / Prozessschnittstelle sowie die Beziehungen bzw. Interaktionen der Software zur Umgebung zu beschreiben.

Übergreifende Abhängigkeiten
In Figure 3 werden farblich „dependencies" als übergreifende Abhängigkeiten angedeutet. Abhängigkeiten sind ein wesentlicher Aspekt zu Modellierung von Software in einem technischen Kontext. Allerdings wird im gesamten Dokument dieses Thema nicht als Bestandteil der Modellierungstheorie aufgegriffen.

2. Detaillierte Kommentare

Kapitel 1:
Abschnitt Fragmentation and isolation of semantic foundations (Seite 4):
Trotz des Bezugs auf die horizontale Integration sind die Beispiele deutlich in unterschiedlichen Prozessphasen angesiedelt. Use Cases werden insbesondere im Requirementsengineering, Zustandsautomaten in späteren Phasen verwendet. Aus diesem Grund sollten die Beispiele überarbeitet werden. Des Weiteren ist die Integration nicht grundsätzlich unmöglich. So eignen sich Sequenzdiagramme und Zustandsautomaten sehr gut für eine Konsistenzprüfung.

Letzter Absatz (Seite 5):
Hier sollte betont werden, dass sich die Theorie ausschließlich mit der Software beschäftigt, ansonsten ist die Aussage „all relevant views (aspects)" nicht konsistent mit dem übrigen Dokument.

Kapitel 2:
Abschnitt Multi-Functional and Complex Systems (Seite 5):
Hier sollte der Begriff Parallelität fallen.

Abschnitt Interactive and Reactive Systems (Seite 5):
Für den unbedarften Leser wäre hier sicherlich ein Beispiel zur Unterscheidung von reaktiven und interaktiven Systemen sinnvoll.

Abschnitte Discrete Controllers and Continuous Physical Processes und Continuous/Hybrid Modeling (Seite 6):
Der erste Abschnitt besagt dass kontinuierliche, physikalische Prozesse mit der vorgestellten Theorie modelliert werden können, wobei der zweite Abschnitt dem widerspricht und behauptet, dass die Theorie keine kontinuierliche Zeit betrachtet. Das auf Seite 12 vorgestellte Zeit-

modell einer diskreten Zeit und die Modellierung von „Channel Histories“ auf Seite 14 unterstützen dabei die Aussage, dass kontinuierliche Vorgänge nicht modelliert werden können.

Kapitel 3:
Abschnitt Modularity and Compositionality (Seite 7):
Hier sollte dargestellt werden, dass sich die Forderung nach Komposition derzeit nur auf das funktionale Verhalten bezieht.
Die Konsequenz, dass aus „correct by construction“ die Notwendigkeit von „property preserving“ folgt, ist aus Sicht der Autoren nicht korrekt. Insofern sollte dieser Abschnitt umformuliert werden.

Abschnitt Interactive and Reactive Systems (Seite 7):
Für den unbedarften Leser wäre hier sicherlich ein Beispiel zur Unterscheidung von reaktiven und interaktiven Systemen sinnvoll.

Kapitel 4:
Abschnitt 4.1 (Seite 9-10)
Der Begriff Service ist aus Sicht des Automatisierungsbereichs problematisch, da er insbesondere die Assoziationen mit Web Services weckt. Des Weiteren ist auch die nachfolgende Definition (Each services specifies a piece of…) bedenklich, da nicht jeder Dienst (komplett) von der Umgebung wahrgenommen werden muss. Bestes Beispiel ist der Switch-Dienst. Ein Dienst der komplett vor der Umgebung verdeckt wäre, ist ein Dienst zur Kommunikation im verteilten System.
Das Beispiel zu Component Networks (Bild 5) ist irreführend, da es ein physikalisches Netzwerk darstellt und damit den Leser zunächst zur Annahme führt, dass Component Networks ausschließlich zur Spezifikation von physikalischen Komponenten dienen. Hier wäre ein anderes Beispiel hilfreicher (oder es wird nur Bild 6 herangezogen).

Abschnitt 4.2 (Seite 11-12):
Zum Teil sind die Kommunikationsparadigmen aufeinander abbildbar. Explizite Nachrichtenbasierte asynchrone Kommunikation mit Speichergröße 1 entspricht beispielsweise der impliziten Kommunikation. Generell ist von einer begrenzten Speichergröße auszugehen, so dass die Annahme der Nicht-Blockade fragwürdig ist.
Die Forderung nach Zeitsynchronisation ist eine sehr harte Anforderung, die häufig nicht notwendig ist.

Abschnitt 4.3
Channel History, S.14: Frage: Können Kanäle unterschiedliche Zeitraten besitzen? Dies erleichtert in der Regel die Modellierung.

Definition 5:
Die Bedingung F(x_1)\neq \emptySet \neq F(x_2) sollte nicht notwendig sein, da F(x_1)=\emptySet \Rightarrow F(x_1↓t)=\emptySet folgen muss. Alles andere ist für eingebettete Systeme nicht sinnvoll (Nachrichten können nicht verschwinden). Insofern fällt dann aber Definition 5 mit Definition 4 zusammen. Das Beispiel 8 ist auch kritisch zu sehen. Wenn zu einem Zeitpunkt t das nicht akzeptierte Zeichen eintrifft, dürfen nur die folgenden Ausgaben undefiniert sein. Für die bisherigen Ausgaben muss natürlich eine Ausgabe entsprechend der Spezifikation gelten.

Abschnitt 4.4
Beispiel Seite 20, siehe Kommentar Komposition in Bezug auf zeitliches Verhalten.

Definition 12, Seite 12: Frage: stellt dies eine eindeutige Definition dar?

Kapitel 5
Dieses Kapitel hängt im Moment etwas in der Luft. Insbesondere ist unklar, wie man von „stream processing functions“ effizient zu „state charts“ kommen kann. Vielmehr scheinen sich beide Konzepte zu ergänzen, da Zustandsautomaten noch Konzepte der alternativen Ausführung bieten, die in den vorherigen Kapiteln fehlte.

3. Rechtschreibkorrekturen und sonstige Fehler

- Kapitel 4, 1. Abschnitt: These views ~~are~~ address
- Inkonsistenz zwischen Abbildung 6 und Abbildung 4: AND und OR-Service haben in einem Fall 2, im anderen Fall 3 Eingänge.
- Definition 8: anstelle von O_1\in H(O_1) müsste es wahrscheinlich y\in H(O_1) heißen

References

[Buc09] Christian Buckl. Kommentare zu der Modellierungstheorie aus Sicht der Anwendungsdomäne Automatisierungstechnik, 2009.

[Fri09] Martin Fritzsche. Kommentare zu der Modellierungstheorie aus Sicht der Anwendungsdomäne Energie, 2009.

[HHR09] Alexander Harhurin, Judith Hartmann, and Daniel Ratiu. SPES 2020 Deliverable D1.1.A-1: Motivation and Formal Foundations of a Comprehensive Modeling Theory for Embedded Systems. Technical report, Technische Universität München, 2009.

[Hol09] Jörg Holtmann. Kommentare zu der Modellierungstheorie aus Sicht der Anwendungsdomäne Automotive, 2009.

[spe] SPES-Wiki. https://spes.informatik.tu-muenchen.de/index.php/ZP-AP_1.1._Liste_von_%C3%84nderungsw%C3%BCnschen_an_die_Modellierungstheorie.