

Verifikation von Spezifikationsmodellen mit Intervall-Petri-Netzen

Vesselka Duridanova

Thorsten Hummel

Olga Fengler

Wolfgang Fengler

Technische Universität Ilmenau

Institut für Theoretische und Technische Informatik

Fachgebiet Rechnerarchitektur

email: wolfgang.fengler@tu-ilmenau.de



Gliederung

- 1. Motivation**
- 2. Message Sequence Charts / Intervall-Petri-Netze**
- 3. Formale Analyse des Intervall-Petri-Netzes**
- 4. Anwendungsbeispiel**
- 5. Zusammenfassung / Ausblick**

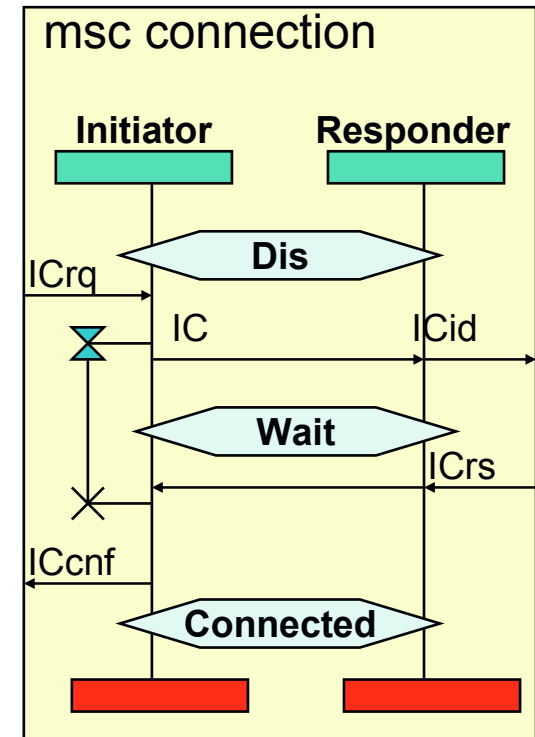
Motivation

Eingebettete Systeme:

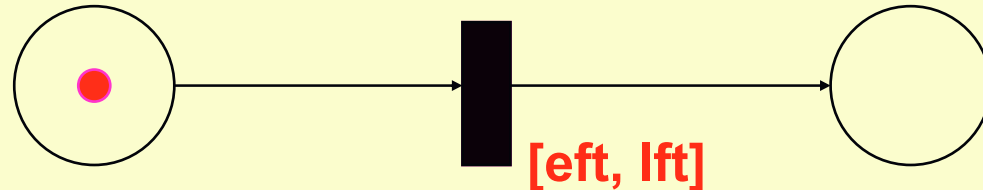
- Komplexe Hard- und Softwaresysteme
 - Einhaltung harter zeitlicher Restriktionen
 - Berücksichtigung in frühen Entwurfsphasen
 - Konsequente Verifikation in allen Entwurfsphasen
- Erhöhung der Entwurfsqualität,
Verkürzung der Entwurfszeiten

Message Sequence Charts (MSC)

- Beschreibung von:
 - Funktionalität
 - Verhalten
 - Interaktion durch Nachrichten
- Nachrichtenaustausch meist asynchron
- Instanzen:
 - Hardware
 - Softwareprozesse
 - Benutzer
- Darstellungsdimensionen:
 - Vertikal – klassische Zeitachse
 - Horizontal – Spezifizierung der betrachteten Objekte
- Zeitbehaftete MSCs
 - Zeitkonstrukte mit Nachrichten verbunden
 - Zeitbedingungen neben Lebenslinie
 - Zeitintervalle möglich



Intervall-Petri-Netze



- Jeder Transition wird ein Zeitintervall zugeordnet, innerhalb dessen die Transition feuern kann

t[a,b] $0 \leq \text{eft} \leq \text{lft}$ **eft** → earliest firing time (Startzeit)

lft → latest firing time (Stoppzeit)

- Transition schaltfähig, wenn Startzeit erreicht
- Wenn Stoppzeit erreicht wird, gerät die Transition in Schaltzwang

Analyse von Pfaden im Erreichbarkeitsraum

- ▣ parametrische Darstellung von Transitionssequenzen
- ▣ Berechnung von optimalen Zeitparametern

Überführung von MSC in Intervall-Petri-Netze

■ Vorteile

- Unterstützung von modularer Strukturierung
- Vereinfachung von Hierarchiebildung
- Vereinfachung von Analysetechniken
- automatische Überführung und Analyse von Zeitanforderungen in MSC ohne detaillierte Kenntnisse des Petri-Netz-Formalismus

■ Nachteile

- IPN komplexer und unübersichtlicher

Überführung von MSC in Intervall-Petri-Netze

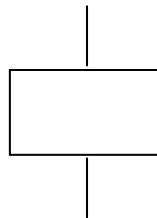
- Ausgangspunkt sind MSCs im Textformat nach Z.120
- Umwandlung durch einen Parser
- Übersetzung der MSC-Grundelemente in IPN-Teilnetze
- Bearbeitung der textlichen Darstellung des MSC in mehreren Durchgängen

InstanzName: instance Name;

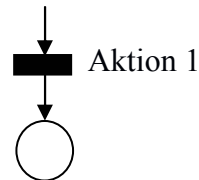
Instanz 1



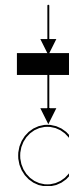
InstanzName: action 'Name';



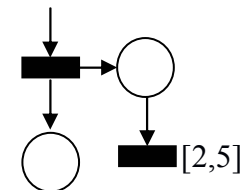
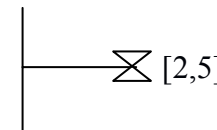
Aktion 1



InstanzName: endinstance;



InstanzName: set Name, Zeitwert;



Analyse von Intervall-Petri-Netzen

- Verfolgung der Pfade im Erreichbarkeitsgraphen
- Analyse von Prozesseigenschaften mit vertretbarem Rechenaufwand
- Gegeben:
 - Anfangszustand
 - Transitionssequenz
- Zwei Schritte:
 - Berechnung der Folgezustände ohne Berücksichtigung der Zeitintervalle
 - Bildung von Zustandsklassen unter Berücksichtigung der Transitionsintervalle

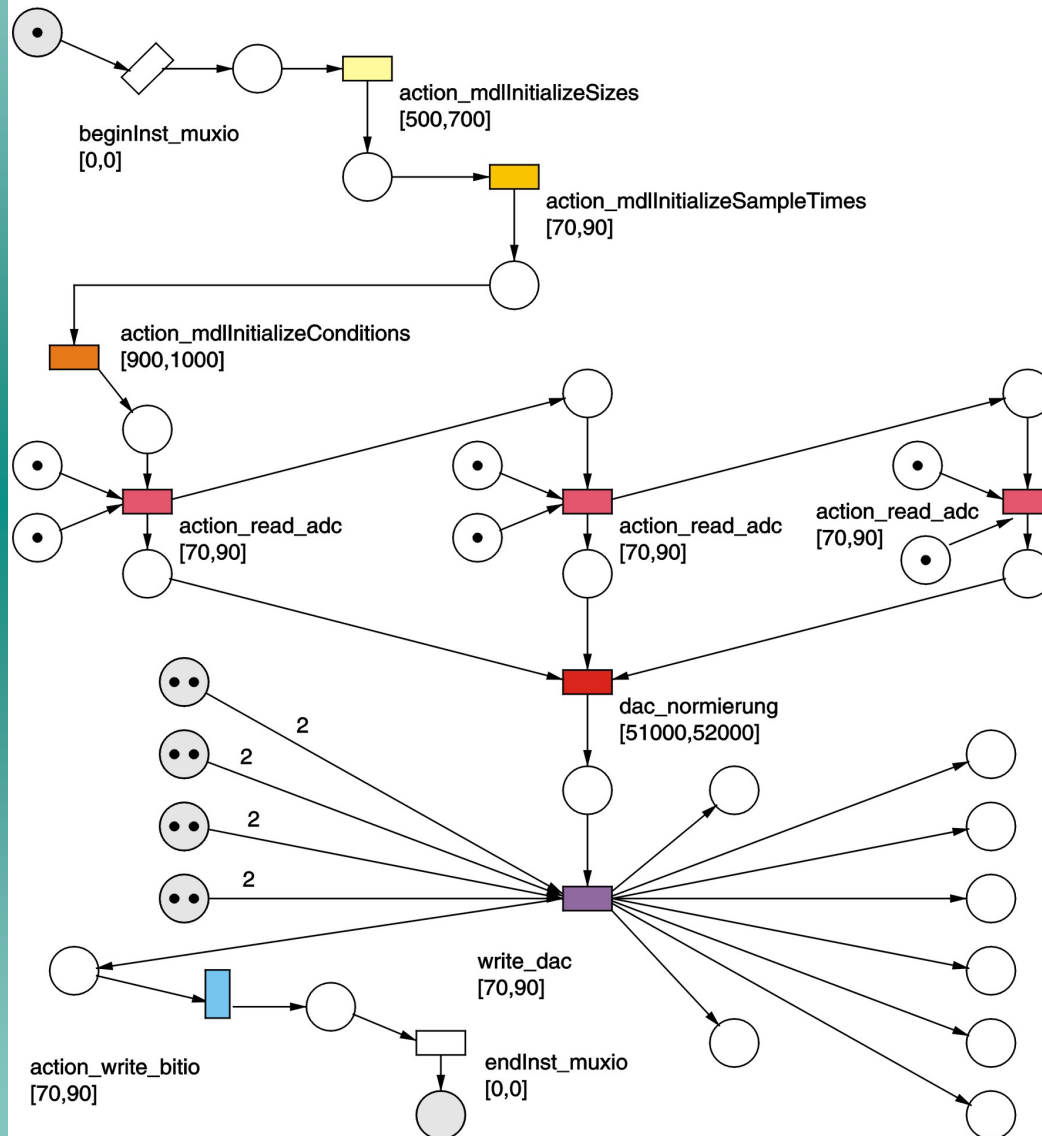
Zustandsklasse: Folge von Zuständen und Zustandsübergängen

Überprüfung von Zeitanforderungen mit Intervall-Petri-Netzen

- σ mit den angegebenen Zeiten ausführbar
- σ erfüllt angegebene Zeiteinschränkungen (deadline-Bestimmung)
- kürzeste und längste Dauer (worst-case-Bestimmung)
- Berechnung von Zeitintervallen für die Transitionen, so dass eine σ ausführbar ist
- Berechnung von Zeitintervallen für die Transitionen, so dass ein bestimmter Zustand (lifelock) nicht erreicht werden kann

$\sigma \rightarrow$ Transitionssequenz

Analyse von Pfaden in einem IPN



▢ feste Zeiten

✓ Bestimmen, ob die Dauer einer Transitionsequenz unter einer Zeitmarke bleibt (deadline)

✓ Bestimmung der kürzesten und längsten Dauer einer Transitionsequenz (worst case Werte)

▢ variable Zeiten

✓ Bestimmung von optimalen Werten für die noch nicht festgelegten Zeiten, so dass die gesamte Transitionsequenz unter einer Zeitmarke bleibt.

worst case [52820, 54240]

Toolunterstützung

▪ Erweiterte Version von VisualObjectNet++

The screenshot displays the Visual Object Net ++ Evaluation Version 2.7a software interface. The main window shows a Petri net diagram with places P1, P2, P3, P4, and P5, and transitions T1, T2, and T3. The diagram includes initial markings and transition labels like [0,1], [1,2], and [1,1].

The **Intervall-Petrinetz-Analyse** dialog box is open, showing a table of transition firing times:

Transition	Value
1: ALT Begin	0
2: T23	0
3: T24	0
4: T25	0
5: T26	6
6: T27	0
7: dimin	0
8: T29	0
9: T30	92
10: T31	0
11: T32	0
12: Limit_Mux	0
13: T44	0

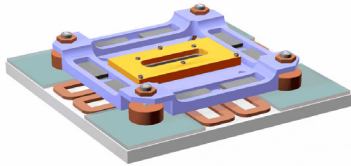
The dialog also shows a table for **Grenzwerte** (Limits):

Maximum	Minimum
98	98

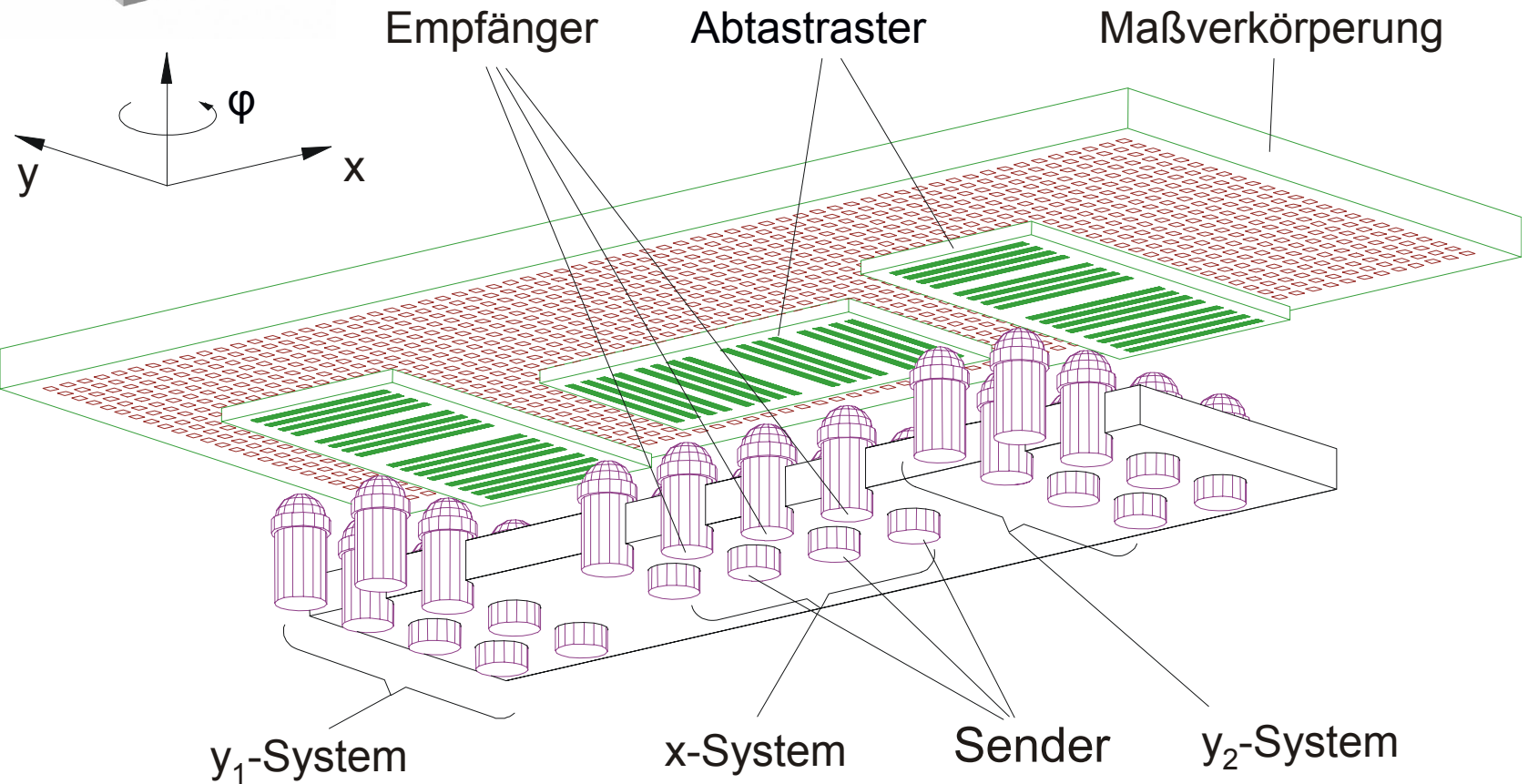
Buttons include **Schaltzeiten übernehmen**, **Schaltzeiten: zufällig**, **Maximum - Minimum - Analyse**, **Überprüfe gegebene Schaltzeit**, **Transition hinzufügen**, **Position optimieren**, **Finde Transitionssequenzen**, **Maximale Anzahl von Sequenzen finden** (set to 1000), **Maximale Länge einer Sequenz** (set to 1000), and **Alle Sequenzen lösen**.

The **Simulation** dialog box is also open, showing options for **Show** (Enabled D-Trans., Enabled C-Trans., Conflict Groups, **Intervall-PN Analyse**) and a **Ready** status with **Time : 0**.

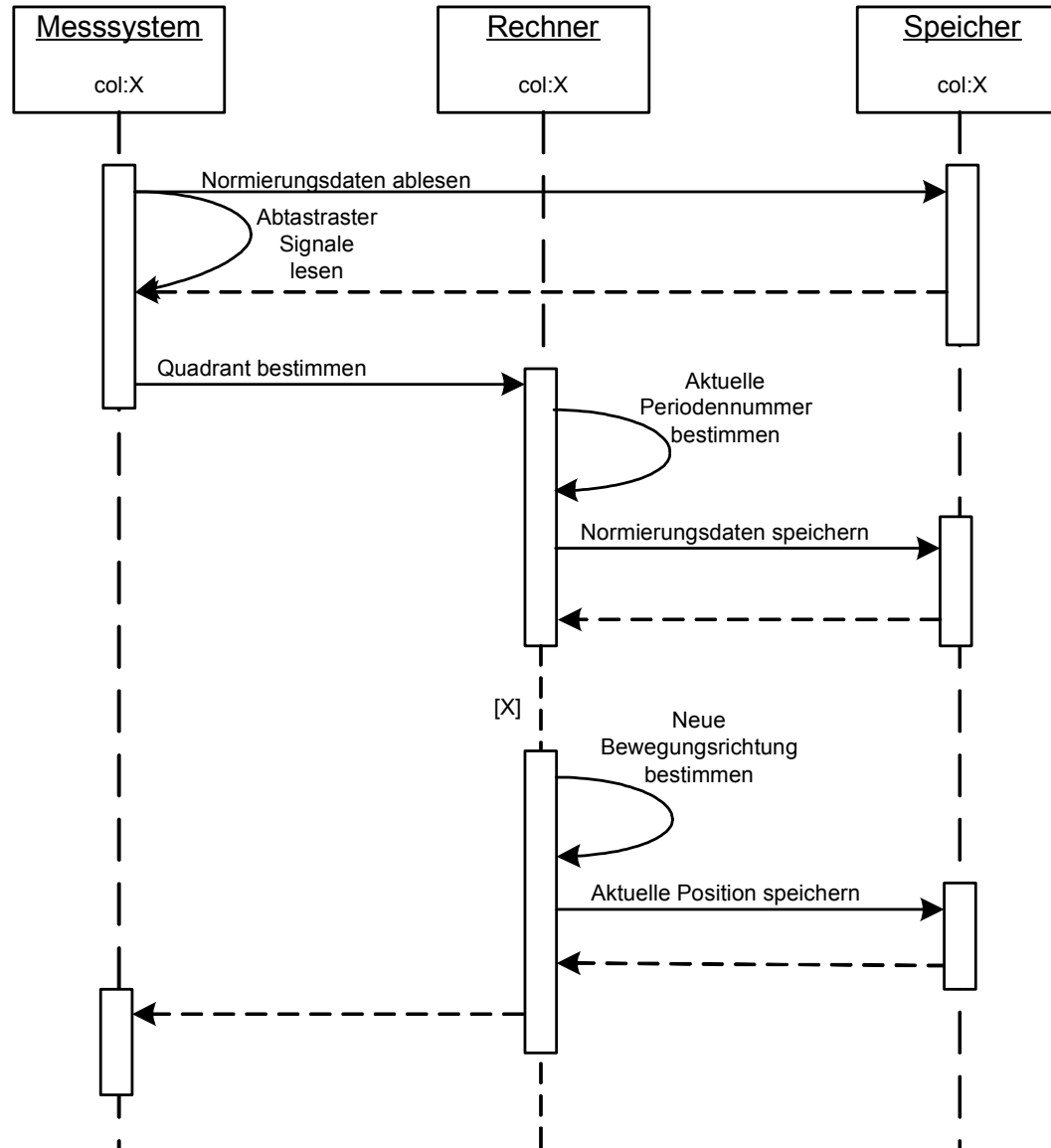
Anwendungsbeispiel



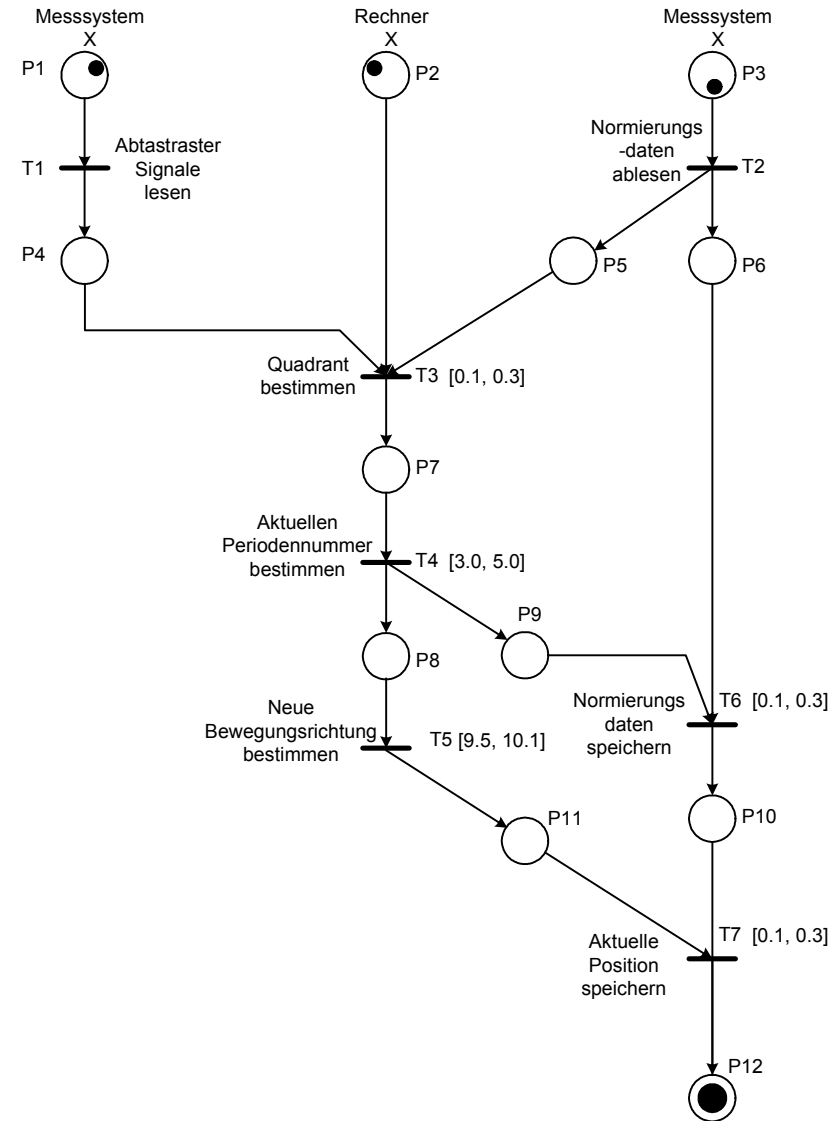
Mehrkoordinatenmesssystem



Message Sequence Chart des Beispielsystems



Intervall-Petri-Netz des Beispielsystems



Zusammenfassung - Ausblick

- Frühzeitige Kontrolle der Einhaltung zeitlicher Eigenschaften komplexer eingebetteter Systeme
- Erweiterung des Entwurfswerkzeuges „VisualObjectNet++“
 - Automatische Umwandlung von MSCs in Intervall-Petri-Netze
 - Formale Analyse von Intervall-Petri-Netzen
- Formale Verifikation von SoC-Entwürfen auf Systemebene