

Entwurfsverfahren für eingebettete Systeme zum Einsatz in der Mechatronik

Wolfgang Fengler, Bernd Däne

Technische Universität Ilmenau

wolfgang.fengler@tu-ilmenau.de

DFG-Schwerpunktprogramm 1040

Informatik 2005 - Bonn 22.09.2005



Gliederung

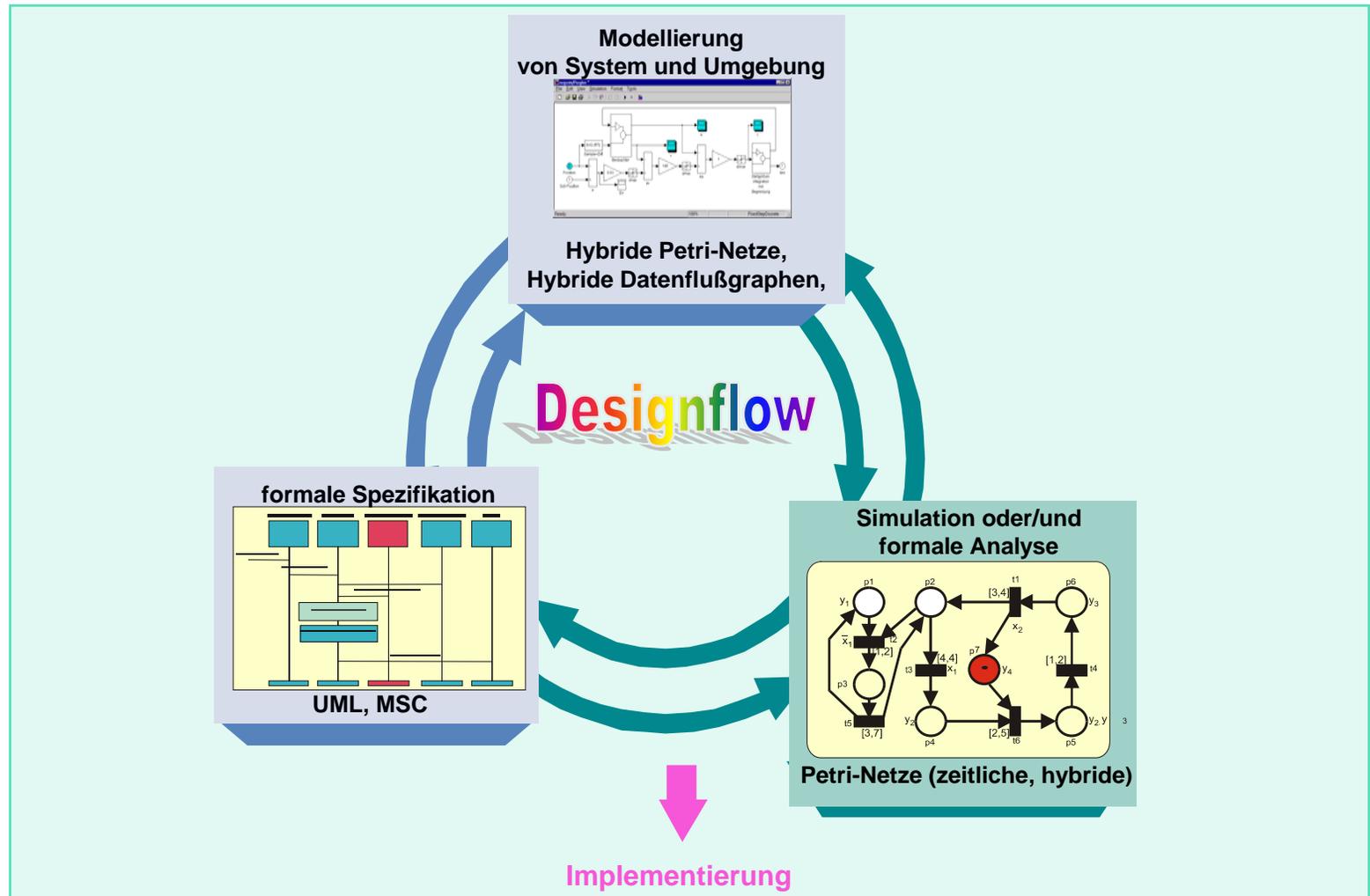
1. Motivation
2. Formale Spezifikation und Analyse
 - Message Sequence Charts (MSC)
 - Colored Message Sequence Charts (CMSC)
 - Intervall-Petri-Netze (IPN)
 - Hybride Petri-Netze
3. Ansätze zur Implementierung
4. Anwendungsbeispiel
5. Zusammenfassung und Ausblick

1. Motivation

Eingebettete Systeme:

- Komplexe Hard- und Softwaresysteme
- Enthalten häufig Elemente, die unterschiedlichen Zeit- und Signalvorstellungen entsprechen
- Oft harte Echtzeitforderungen
- Heterogene oder hybride Systeme
- Beschreibung durch unterschiedliche Formalismen
 - kontinuierliches Zeitmodell
 - diskrete Ablaufschritte

2. Formale Spezifikation und Analyse



Message Sequence Charts (MSC) (nach Z.120, UML)

Beschreibungskontexte:

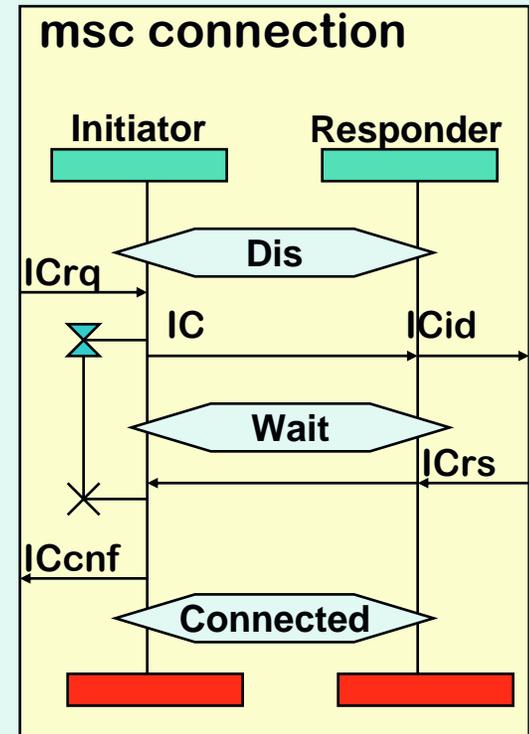
- grafische Repräsentation
- textuelle Notation

Anwendungsphasen:

- Spezifikationsphase
- Entwurfsphase – Entwurfsmuster für reaktive Systeme
- Implementierungsphase – Beschreibung von Testfällen
- Dokumentation

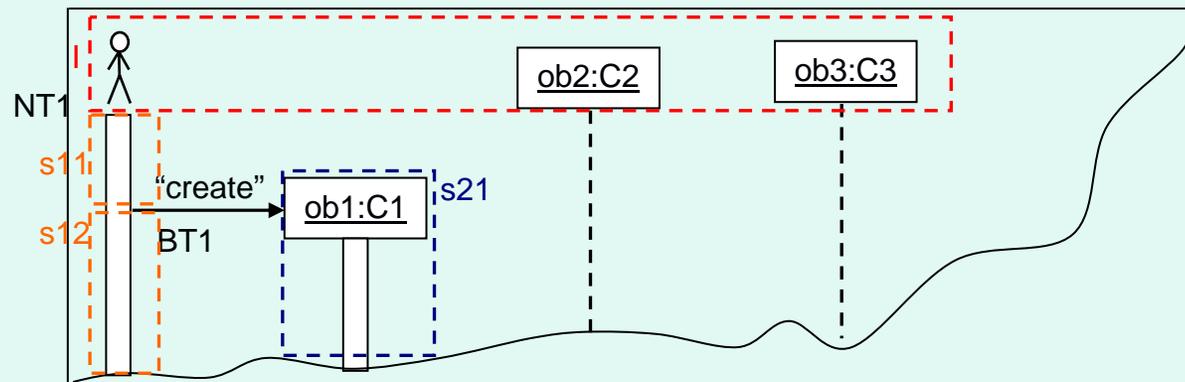
Anwendungsfeld:

- Nachrichtenaustausch zwischen nebenläufigen (Sub)systemen
- Darstellung von Soll- und Fehlverhalten (Good und Bad MSC)
- Darstellung von komplexen Systemen (HMSC/BMSC)



Message Sequence Charts (MSC)

Zustände in MSC



Transitionen in MSC

Normaltransition

Nachricht1()
→

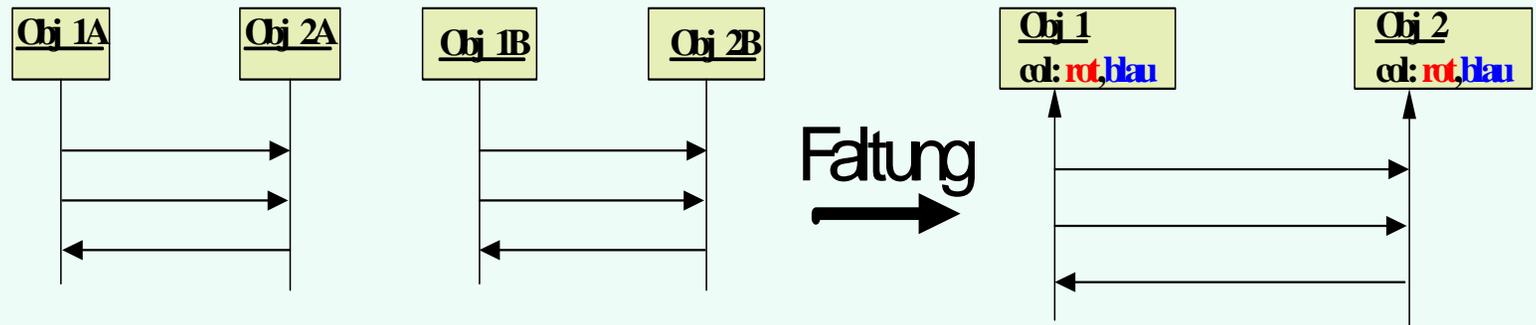
-
-
-

Colored Message Sequence Charts

„Faltung“ :

Überlagern von Instanzen ähnlichen Inhaltes und ähnlicher Struktur

Zusammenfassung mehrerer Diagramme:



Obj iA: Obj i,rot

Obj iB: Obj i,blau

Colored Message Sequence Charts

Faltungsmöglichkeiten:

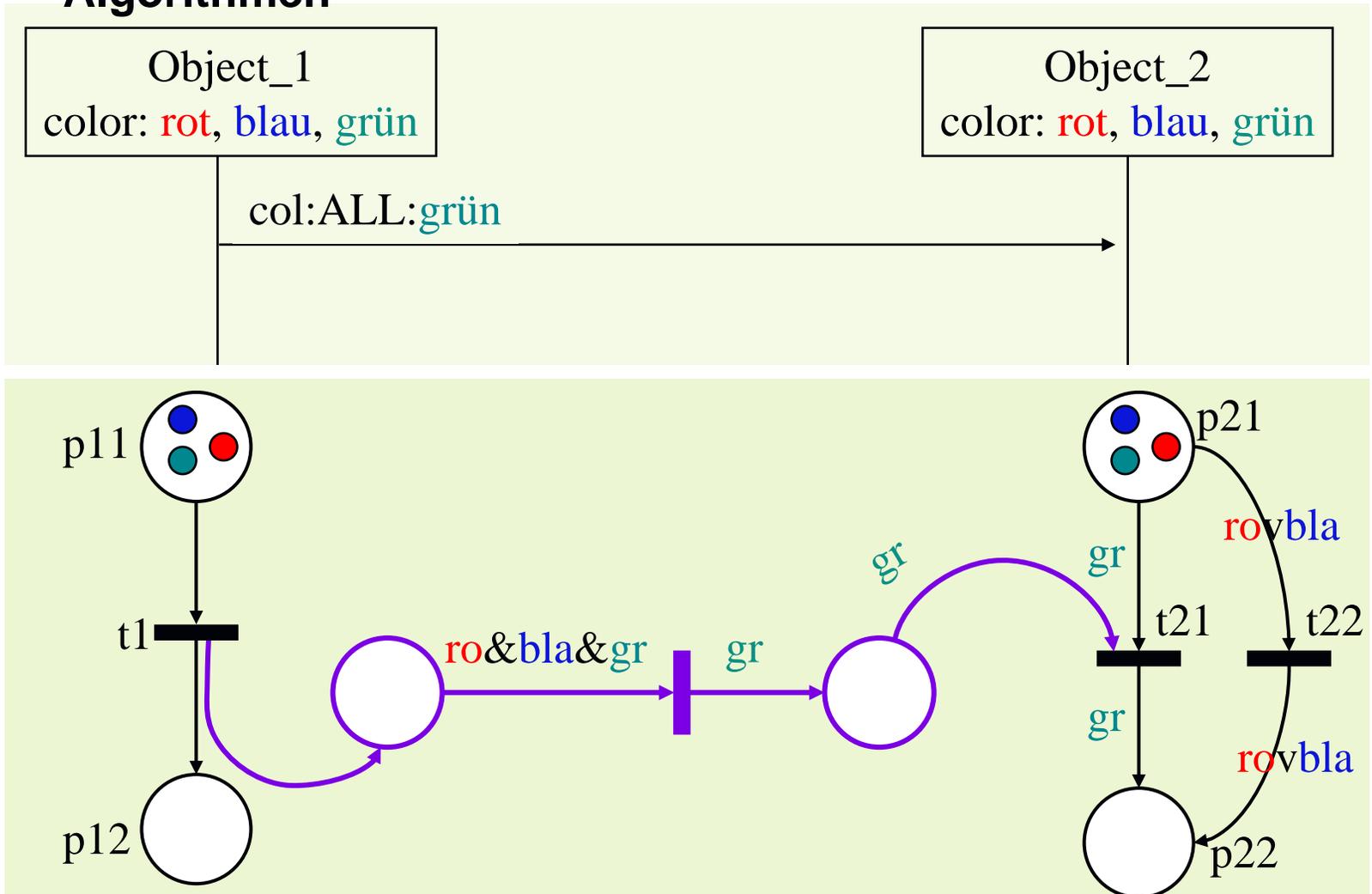
- mehrere Unterklassen
- eine Klasse und deren Unterklassen
- mehrere Objektinstanzen einer Klasse
- mehrere Objektinstanzen verschiedener Unterklassen

Aus der Sicht von UML-Diagrammen folgende Möglichkeiten der Faltung:

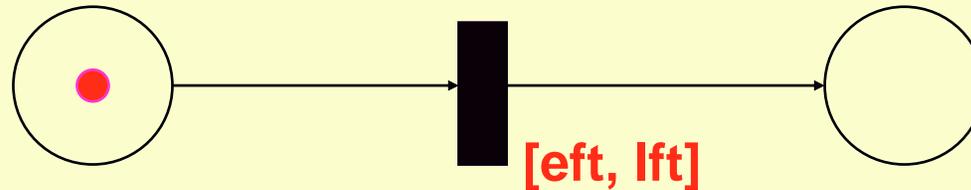
- Zusammenfassung mehrerer Diagramme
- Zusammenfassung mehrerer Instanzen eines Diagrammes

Transformationskonzept CSD => CPN

Transformation von Bausteinen auf Basis von existierenden Algorithmen



Intervall-Petri-Netze



t[a,b] $0 \leq \text{eft} \leq \text{lft}$ **eft** → earliest firing time (0,m)
lft → latest firing time (n,∞) $m=n \vee n>m$

Zustand im Erreichbarkeitsgraphen

Platzbeschreibung → m (Markierung)  **z(m,J)**
Transitionsbeschreibung → J (Zeitvektor)
 $z = ((1,0,3), (2, \#, 0, \#))$

Analyse von Pfaden im Erreichbarkeitsgraphen

- Parametrische Darstellung von Transitionssequenzen
- Berechnung von optimalen Zeitparametern

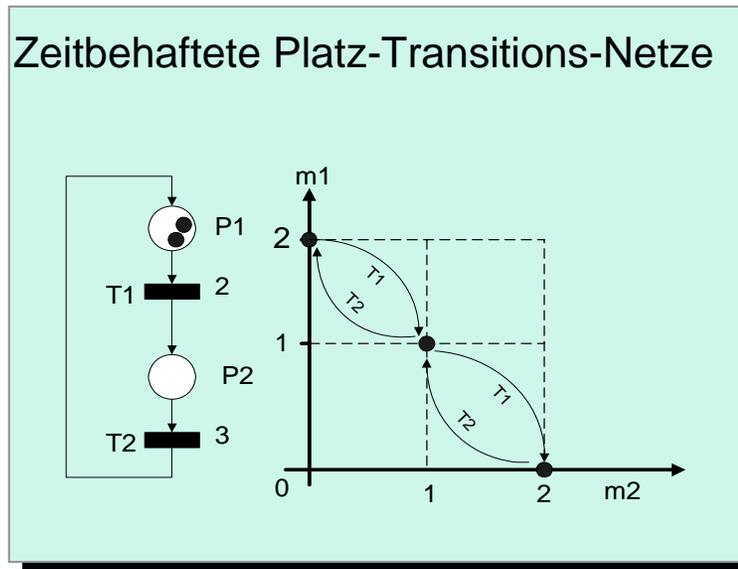
Überprüfung von Zeitanforderungen mit Intervall-Petri-Netzen

- 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

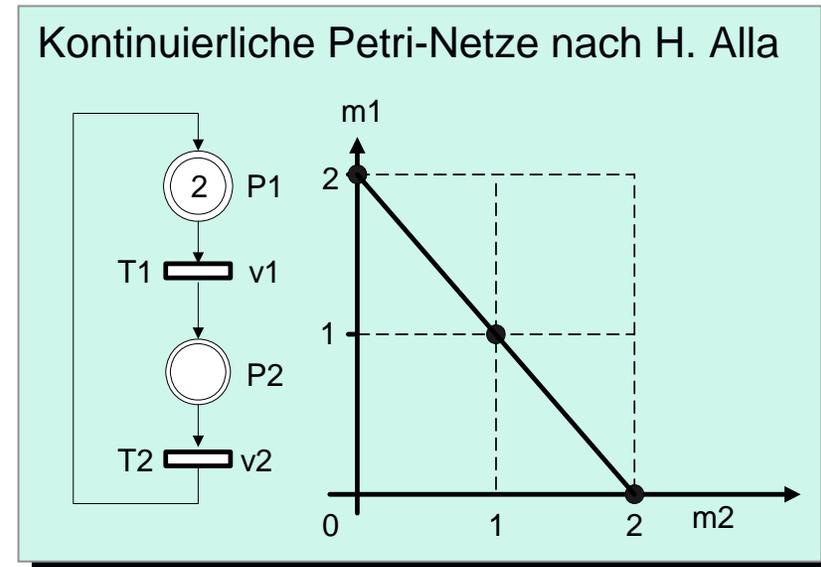
$\sigma \rightarrow$ Transitionssequenz

Hybride Petri-Netze

Modellierung ereignisdiskreter und kontinuierlicher Systeme mit **einem** Beschreibungsmittel, simulierbar



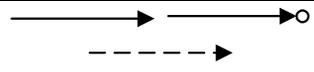
ereignisdiskretes Verhalten



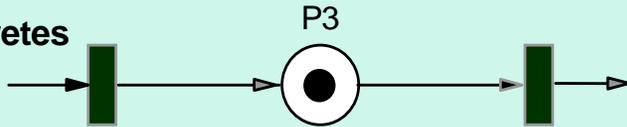
kontinuierliches Verhalten

Hybride Petri-Netze (nach Alla, Drath)

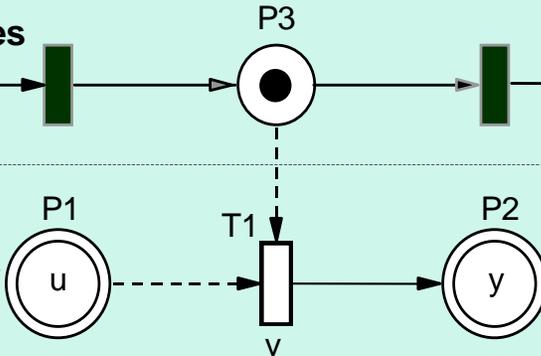
Hybride Petri-Netze - Grundentwurfsmuster

TD 	PD 	TC 	PC 	
diskrete Tran- sition	diskreter Platz	kontinuierliche Transition	kontinuierlicher Platz	Kanten (normale, inhibitorische und Testkanten)

ereignisdiskretes
Teilmodell

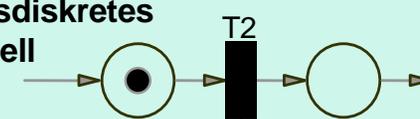


kontinuierliches
Teilmodell

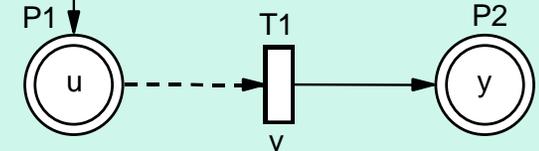


Steuerung kontinuierlicher Prozesse

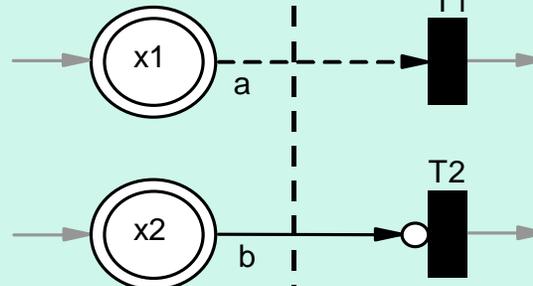
ereignisdiskretes
Teilmodell



kontinuierliches
Teilmodell



Generieren von Sprungfunktionen



kontinuierliches
Teilmodell

ereignisdiskretes
Teilmodell

Generieren von Ereignissen

3. Ansätze zur Implementierung

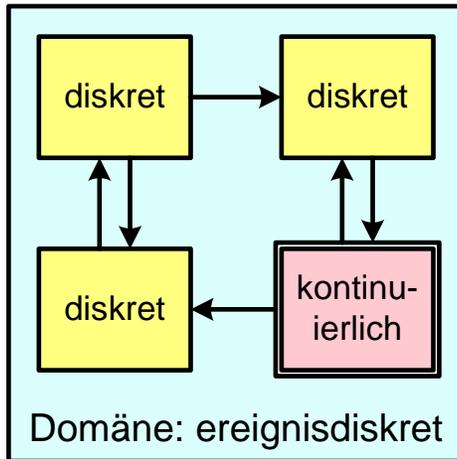
Ziel:

Schließen der Lücke zwischen simuliertem/verifiziertem Modell und Implementierung auf dem Zielsystem

Probleme:

- Unterstützung von Multi-Domain-Modellen
- Koordination konkurrierender Aktivitäten
- Einbindung in Multitask-Realtime-Betriebssysteme

Diskret-kontinuierliche Implementierung



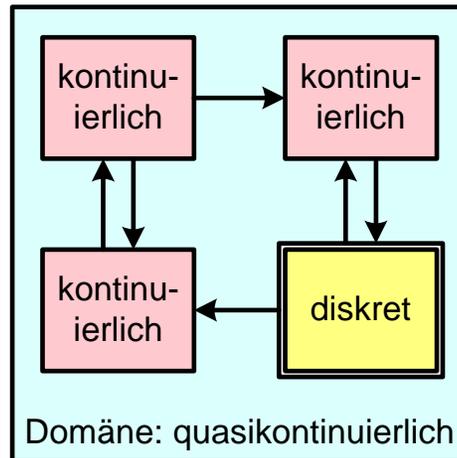
Ereignisdiskrete Domäne:

Blöcke werden durch eintreffende Ereignisse aktiviert

Implementierung als Tasks mit dynamischem Scheduling und Preemption

Quasikontinuierliches Wormhole:

Aktivierung durch ratenmonotone Ereignisfolge



Quasikontinuierliche Domäne:

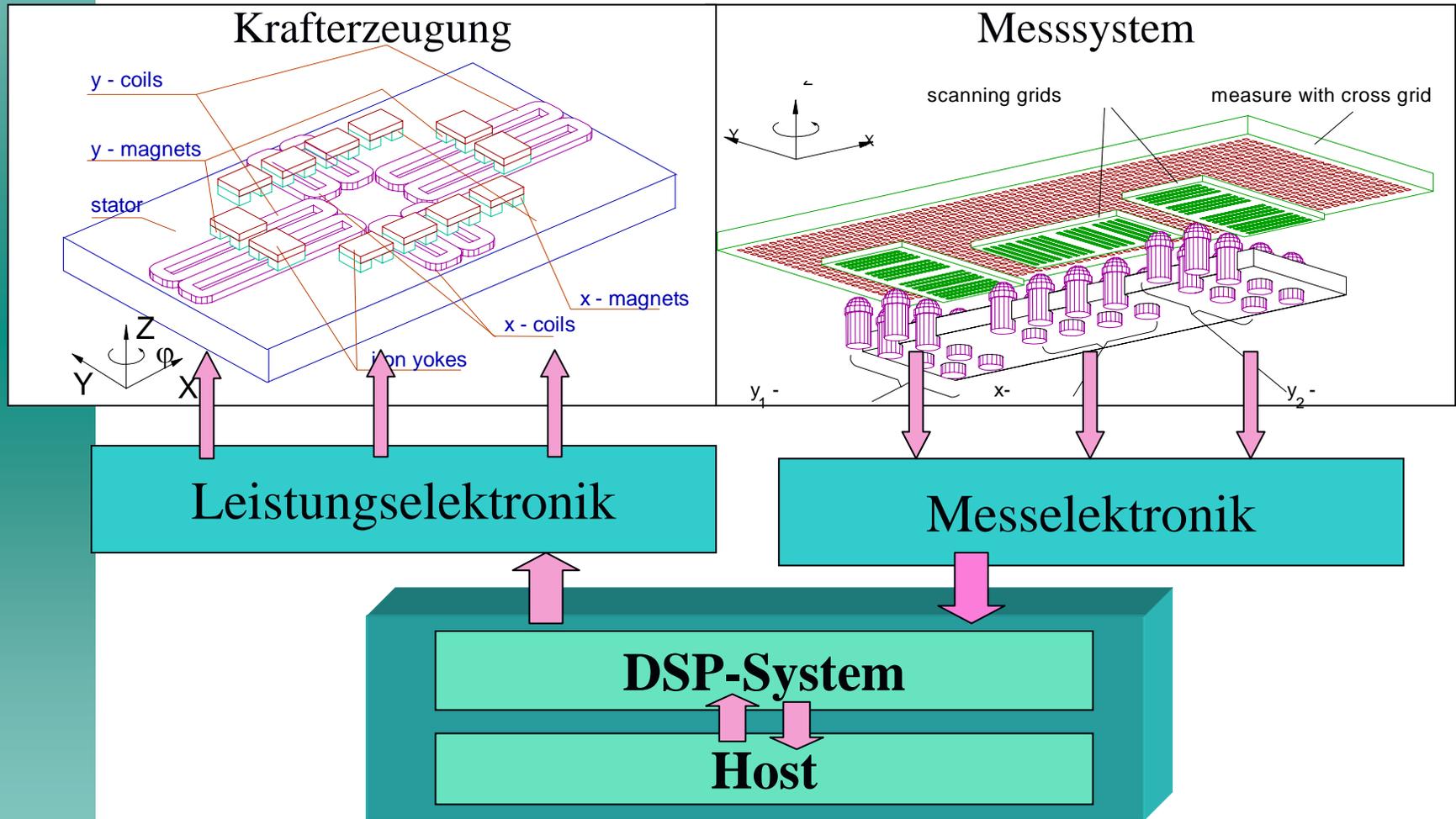
Blöcke werden mit monotoner Rate aktiviert

Implementierung mit ratenmonotonom statischem Scheduling ohne Preemption

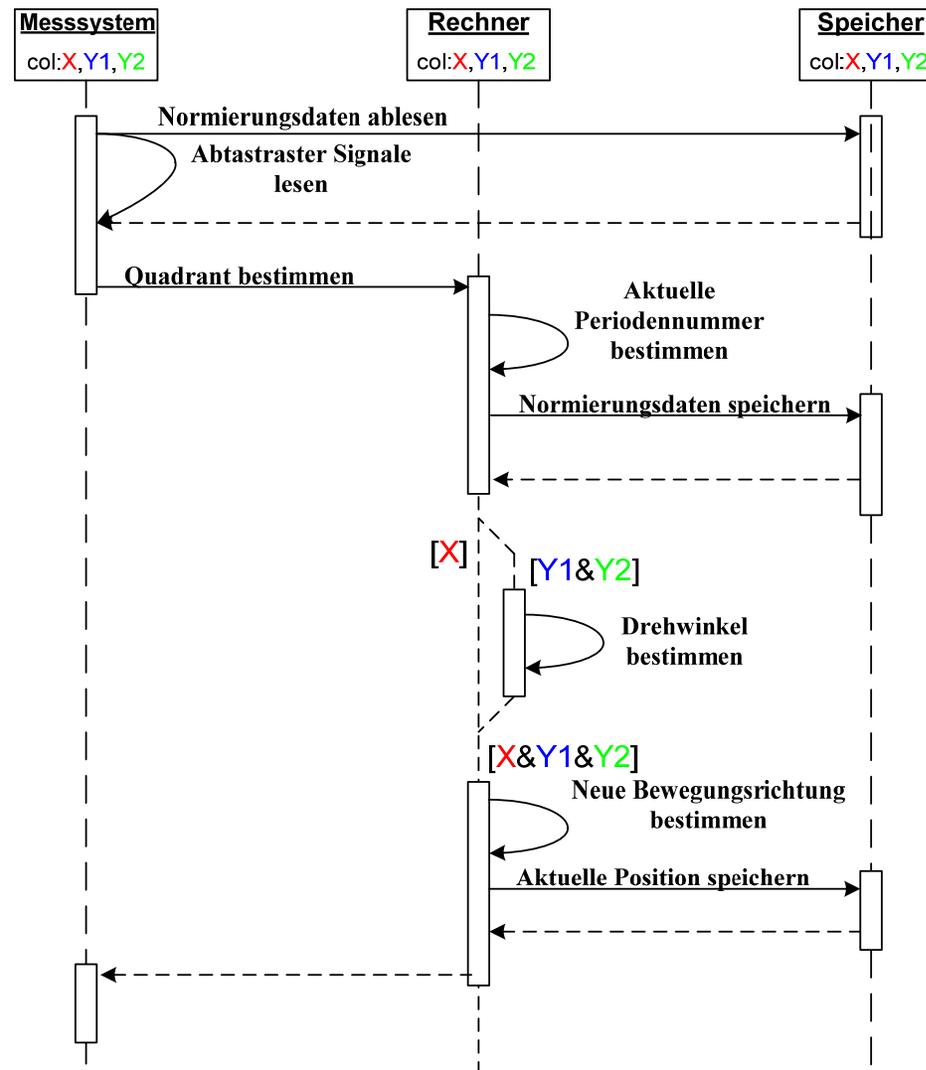
Ereignisdiskretes Wormhole:

Ratenmonotone Aktivierung der Randelemente innerhalb des statischen Schedules, dynamisches Scheduling der inneren Elemente

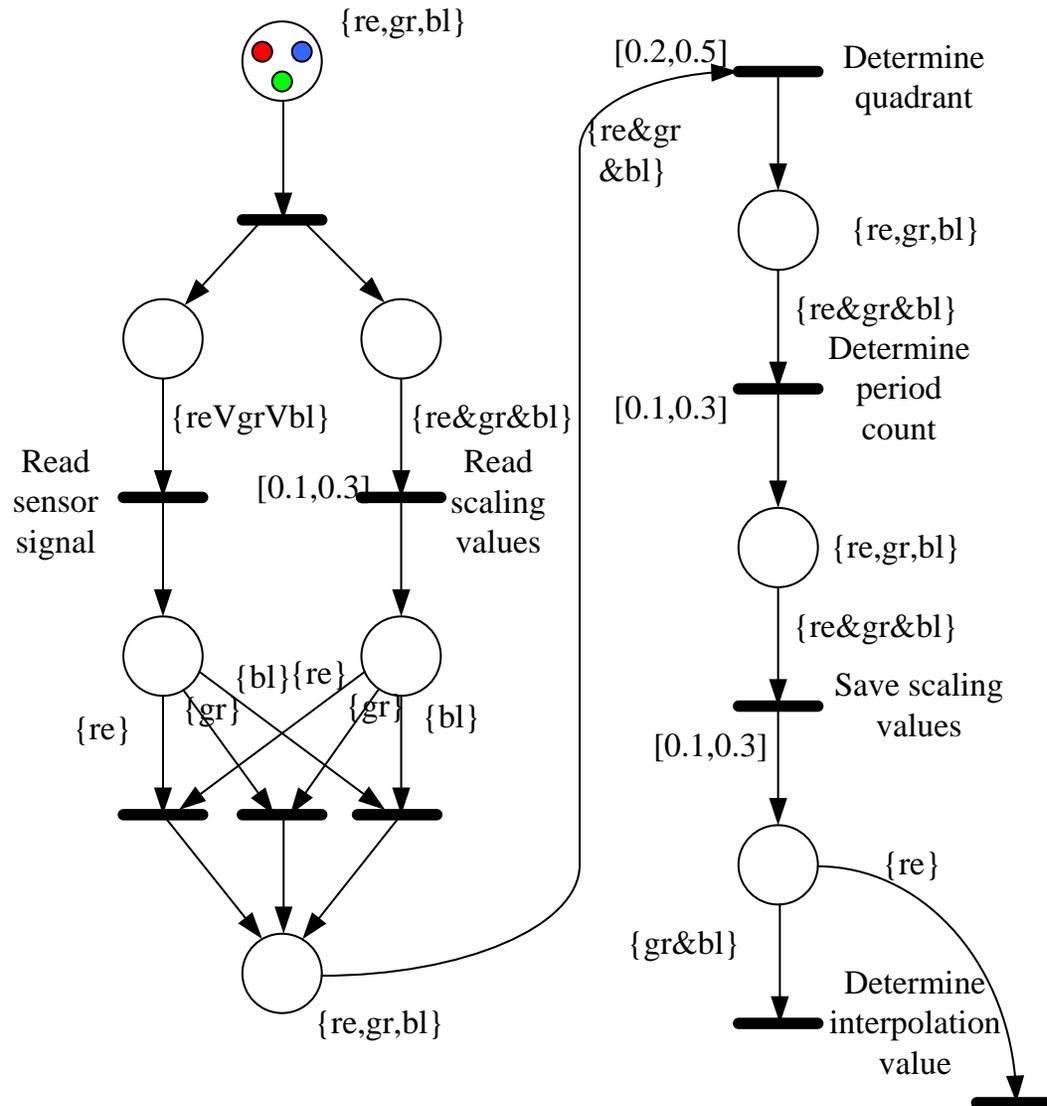
4. Anwendungsbeispiel



Gefärbtes MSC

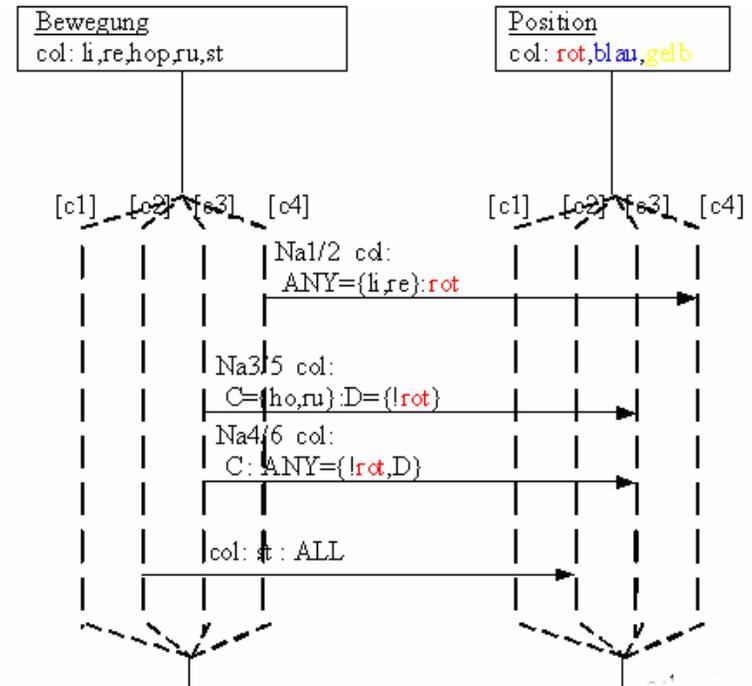
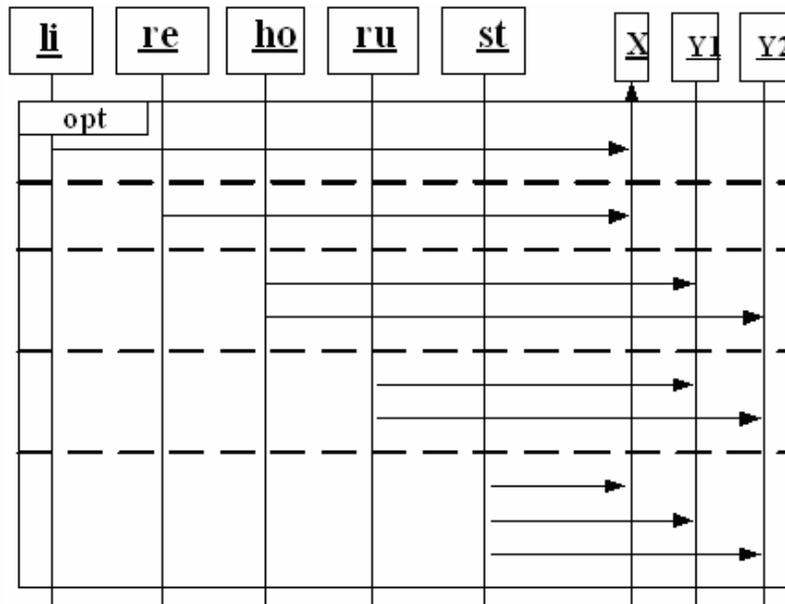


Gefärbtes Zeitintervall-Petri-Netz

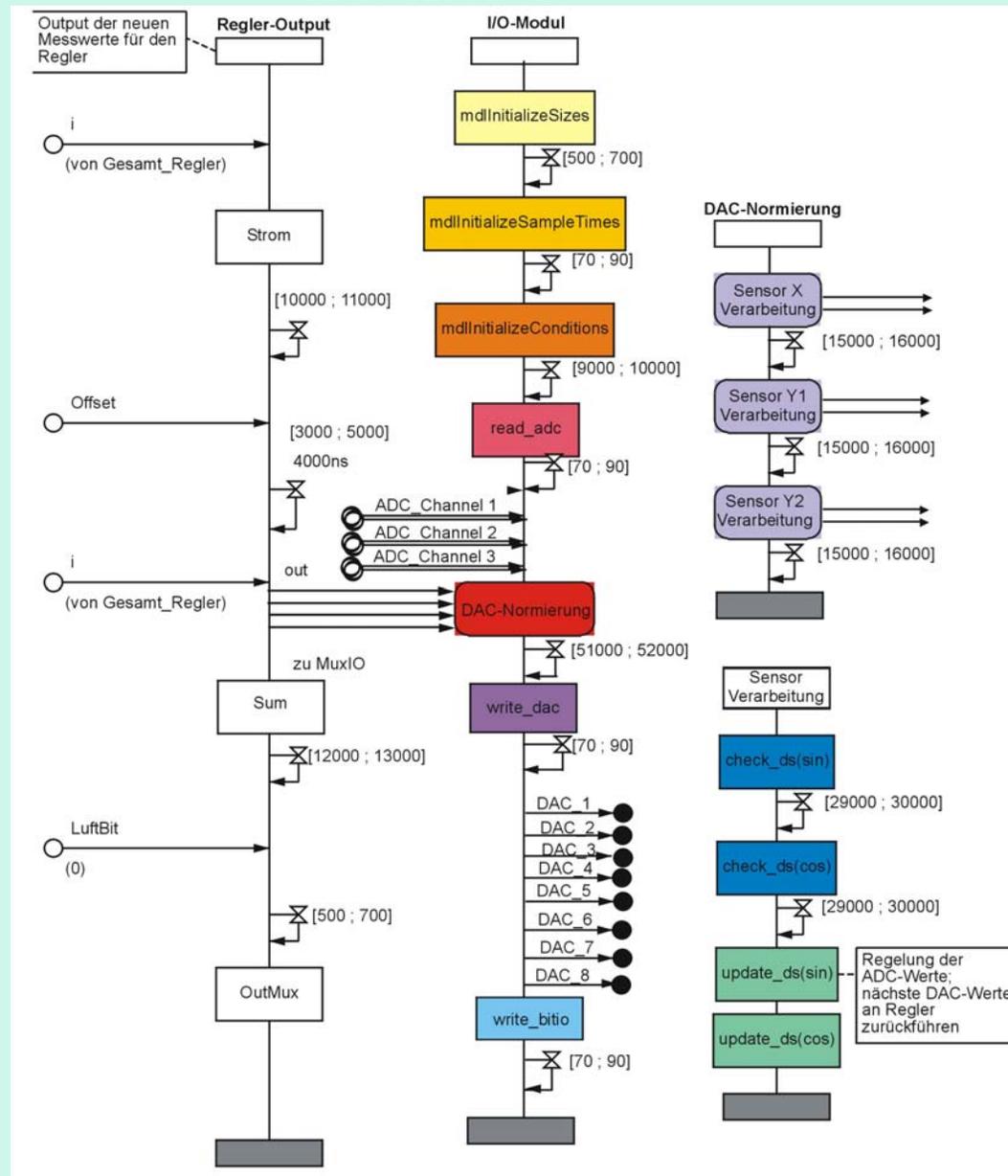


MSC- und CMSC-Ausschnitt

(Bewegung des Messtisches)

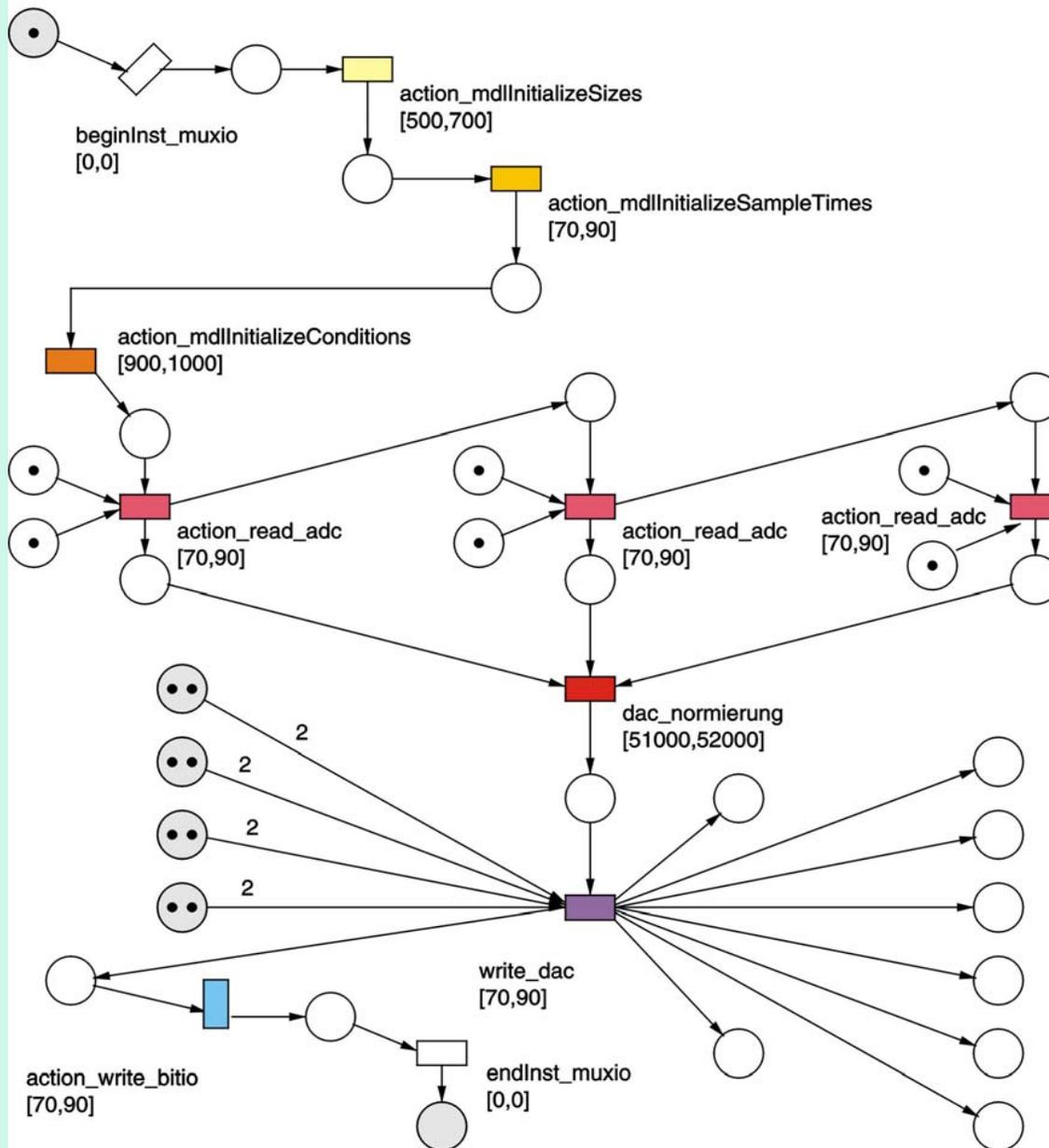


MSC-Modell zum I/O-Modul



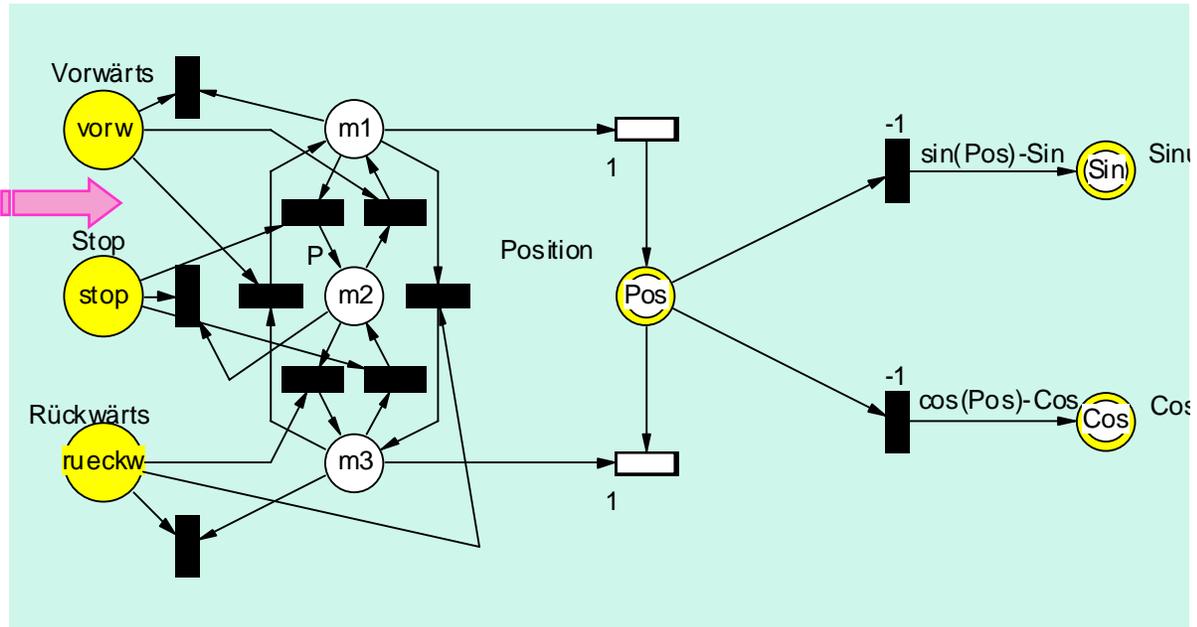
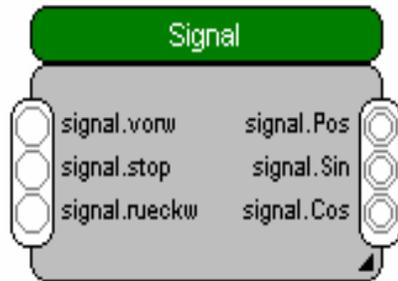
Analyse von Pfaden in einem IPN mit Hilfe von t-INA (Starke)

- feste Zeiten
 - Bestimmung der kürzesten und längsten Dauer einer Transitionssequenz (worst case Werte)
 - Bestimmen, ob die Dauer einer Transitionssequenz unter einer Zeitmarke bleibt (deadline)
- variable Zeiten
 - Bestimmung von optimalen Werten für die noch nicht festgelegten Zeiten, so dass die gesamte Transitionssequenz unter einer Zeitmarke bleibt.



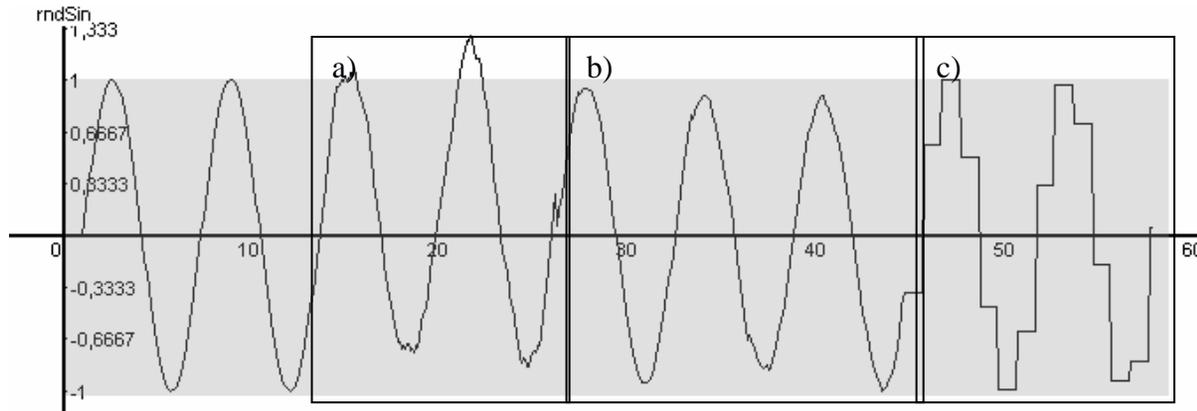
worst case [52820, 54240]

Mehrkoordinatenmesssystem – Objekt „Signal“

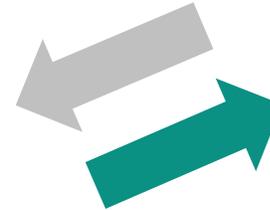
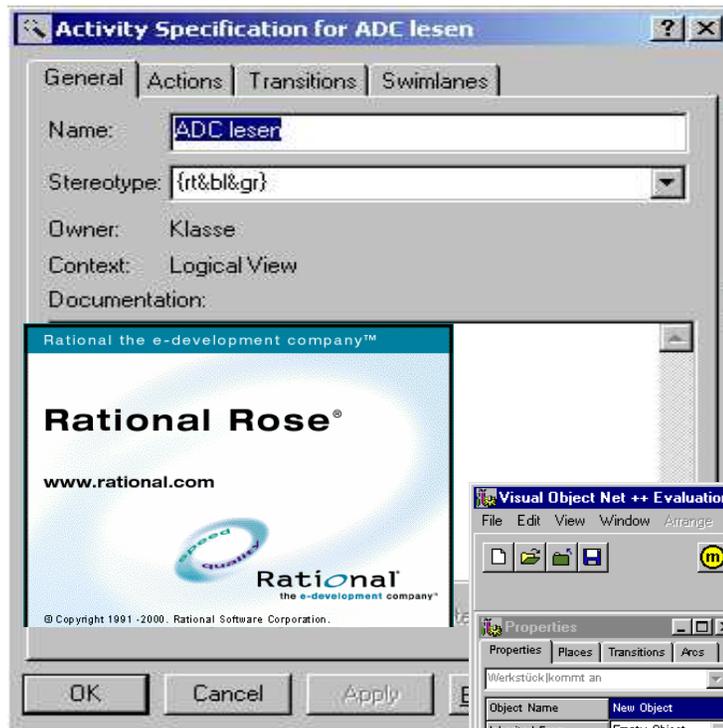


Störungsarten

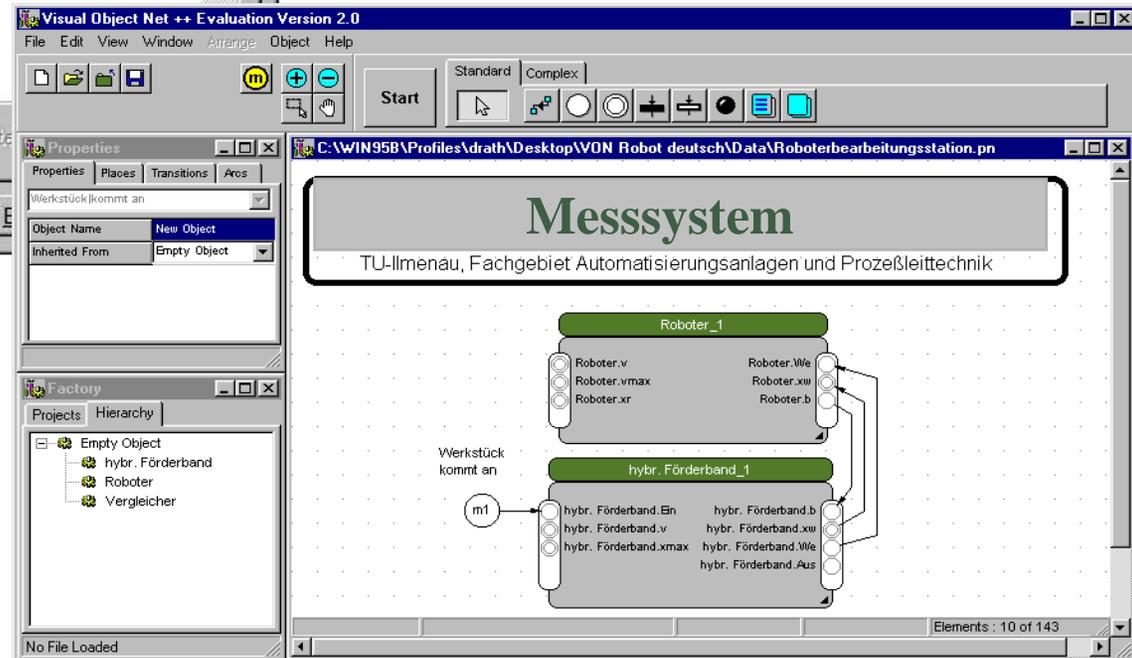
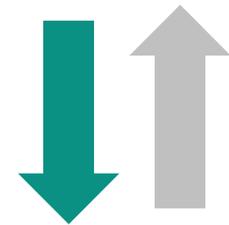
- a) Änderung der Normwerte
- b) Veränderung der Amplitude
- c) Zeitverzögerung



Toolunterstützung



CMSC
Konverter



Integration der Analyse-Methode in den Simulator

Intervall-Petrinetz-Analyse

Datei Optimierungsmethode Ansicht Options Hilfe

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

Grenzwerte
Maximum: 98
Minimum: 98

Schaltzeiten übernehmen
Schaltzeiten: zufällig

Maximum - Minimum - Analyse
Überprüfe gegebene Schaltzeit

Transition hinzufügen
Position optimieren

Finde Transitionssequenzen

Maximale Anzahl von Sequenzen finden: 1000
Maximale Länge einer Sequenz: 1000

3 / 60

Alle Sequenzen lösen

Simulation

Panel Properties Extras

Show

- Enabled D-Trans.
- Enabled C-Trans.
- Conflict Groups
- Intervall-PN Analyse

Ready Time : 0

6. Zusammenfassung und Ausblick

- Entwicklung Gefärbter UML-Diagramme
 - Transformation in Intervall-Petri-Netze
 - Erweiterung der Hybriden Petri-Netze um ein Zeitintervall-Konzept
 - Methodik zur formalen Analyse anhand der Intervall-Petri-Netze
 - Exemplarische Implementierung in Tools
 - Ansätze zur Implementierung diskret/kontinuierlicher Modelle
-
- Entwurfsmethodik und Entwurf hochleistungsfähiger paralleler Signalprozessor-Lösungen für **Nanopositionier- und Nanomessmaschinen** im DFG-Sonderforschungsbereich 622
 - Zeitverifikation für SoC-Lösungen im Clusterforschungsprojekt „**Funktionale Verifikation von Systemen**“