High-Level-Entwurfsmethodik für eingebettete mechatronische Echtzeitsysteme

Vesselka Duridanova,
Bernd Däne, Jürgen Nützel, Wolfgang Fengler
Technische Universität Ilmenau
Institut für Theoretische und Technische Informatik
Fachgebiet Rechnerarchitekturen
e-mail:vesselka, bdaene, nuetzel, wfengler@theoinf.tu-ilmenau.de

TU Ilmenau

Gliederung

1. Entwurfsprozess hybrider Systeme

2. Formale Spezifikation und Analyse

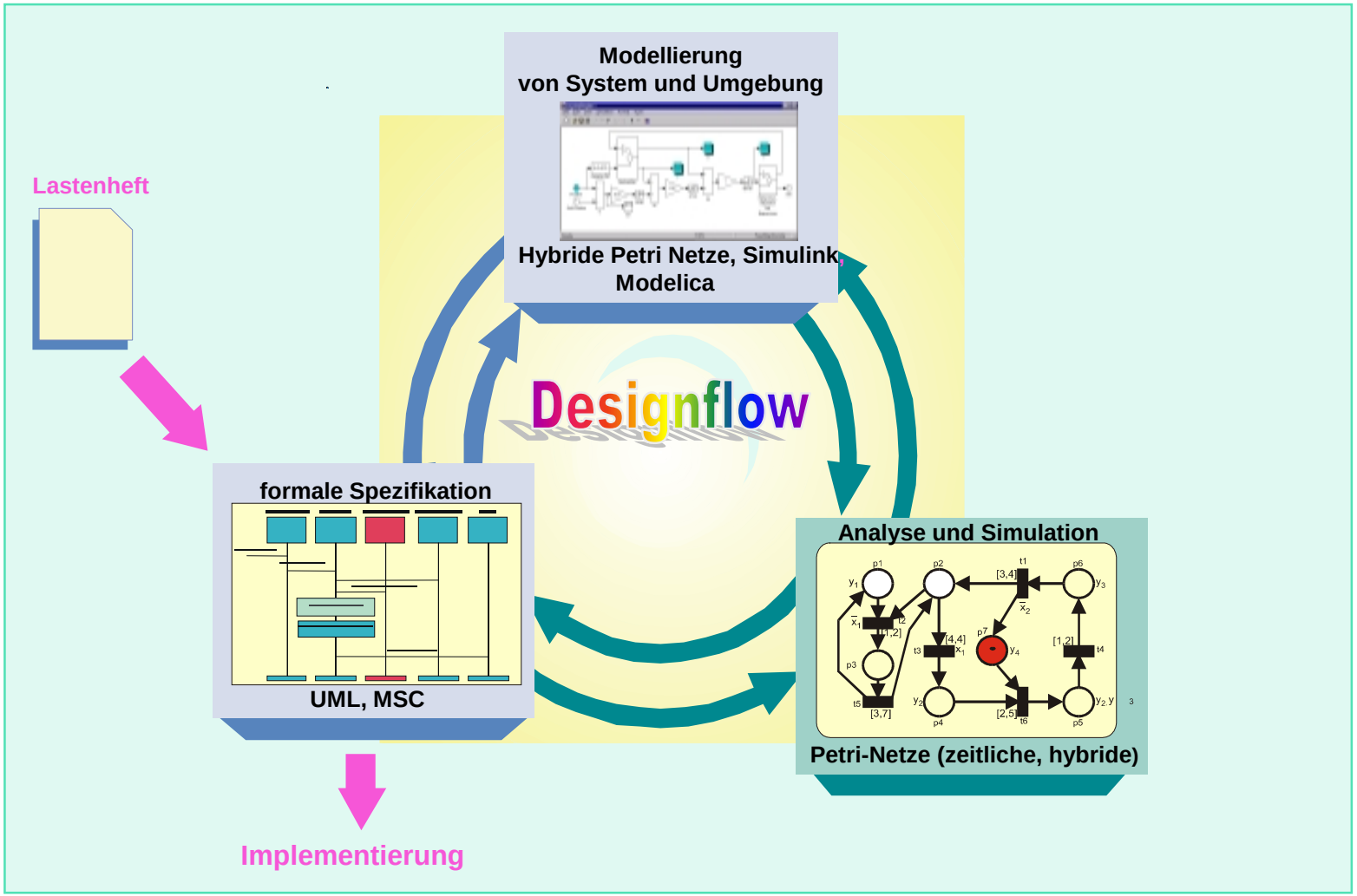
2. 1. Message Sequence Charts (MSC)

2. 2. Intervall-Petri-Netze (IPN)

3. Modellierung und Simulation von hybridem Verhalten

4. Ausblick

Entwurfsprozess hybrider Systeme



Formale Spezifikation = Anforderungsanalyse und Prüfung

Analyse

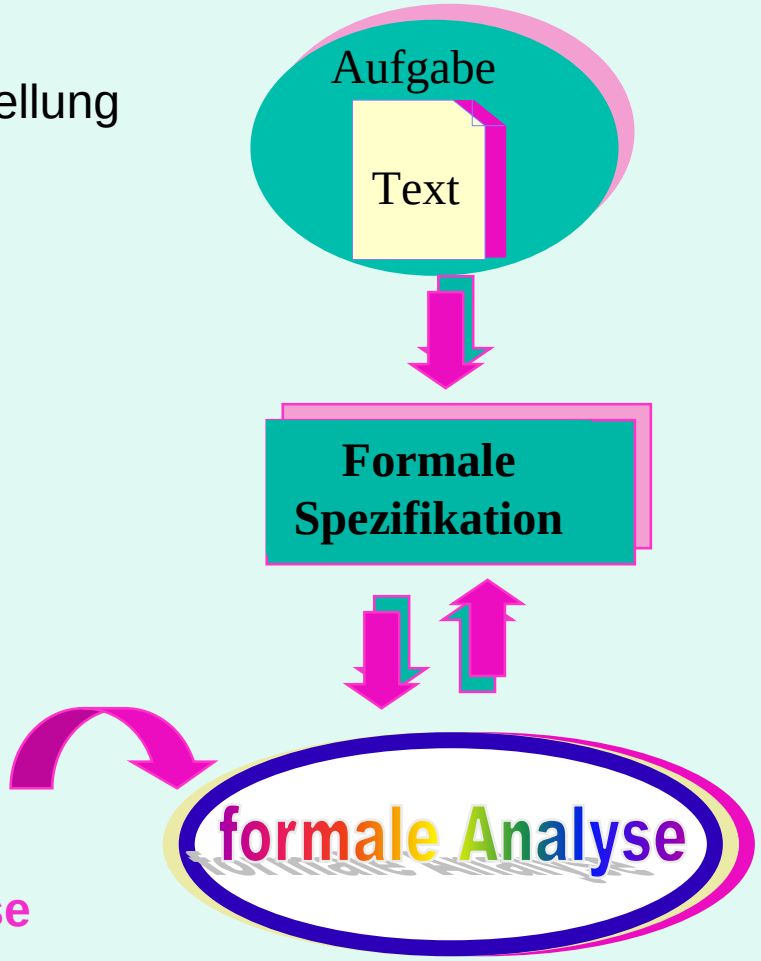
- Formalisierung der Aufgabenstellung
- Systemkomponentenstruktur

Spezifikationssprachen

- UML
  - Use-Case-Diagramme
  - Klassendiagramme
  - Sequenz-Diagramme
- **MSC (HMSC)**

Ziel

- Aufgabe vollständig und widerspruchsfrei formulieren
- Sicherheitsanalyse
  - **Echtzeitanforderungen**
  - **Worst-Case Analyse**
  - **Risiko, Gefahrenanalyse**

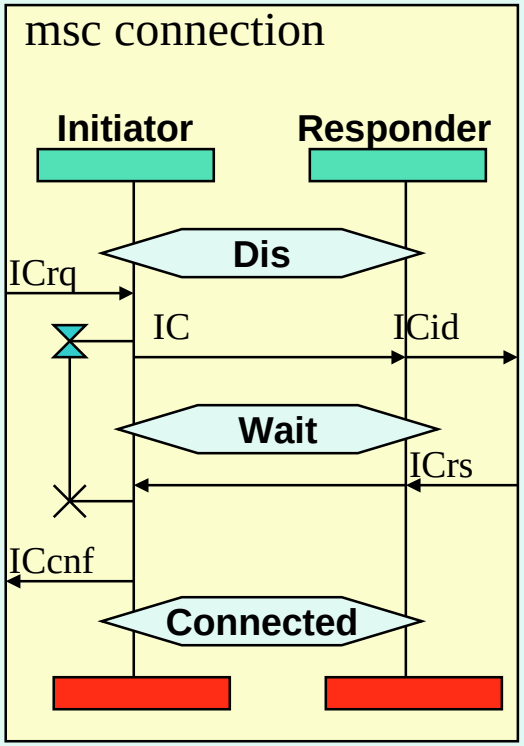


Message Sequence Charts (MSC)

Beschreibungstechniken
- grafische Repräsentation
- textuelle Notation
- algebraische Deskription

Anwendung im Entwurfsprozess
- Requirements Engineering
- Entwurfsphase – Entwurfsmuster
- Implementierungsphase – Beschreibung von Testfällen
- Dokumentation

Spezifikationsmuster
- asynchroner Nachrichtenaustausch zwischen nebenläufigen (Sub)systemen
- Verwendung von Kommunikations-szenarien
- Darstellung von komplexen Systemen (HMSC/BMSC)
- Darstellung von gewünschtem und nichtgewünschtem Verhalten (Good und Bad MSC)



Formale Analyse

Formale Methoden

- Mathematischer Nachweis von (generellen) Systemeigenschaften
- Erfassung des gesamten Zustandsraumes
- Eindeutigkeit garantieren
- frühe Erkennung von Entwicklungsfehlern
- hoher Aufwand bei komplexen Systemen

Toolunterstützung

- automatische Überführung Spezifikation -> formale Methode
  - Nachvollziehbarer Entwicklungsweg
  - keine fundierten Kenntnisse vom Anwender erforderlich
- Wiederverwendbarkeit
- rechnergestützter Korrektheitsnachweis
- Teamunterstützung

Auswahl der geeigneten formalen Analysemethode

Intervall-Petri-Netze (IPN)

$N = (P, T, F, V, m_0, I) \rightarrow$ IPN

(1) P, T, F $\rightarrow$ endliche Mengen mit

$P \cap T = 0,\ P \cup T \neq 0,\ F \subseteq (P \times T) \cup (T \times P)$

$dom(F) \cup cod(F) = P \cup T$ (Netz)

(2) $V : F \rightarrow N^+$ (Kantengewicht)

(3) $m_0 : P \rightarrow N$ (Anfangsmarkierung)

(4) $I: T \rightarrow Q_0 \times (Q_0 \cup \{\infty\})$ und $\forall\ t \in T\quad I_1(t) \leq I_2(t),\ I(t) = (I_1(t), I_2(t))$

$I \rightarrow$ Zeitfunktion von N

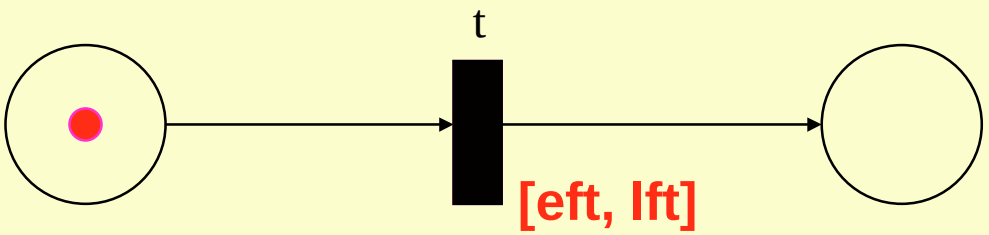
Zustand z = (m,J) **in N** mit $J : T \rightarrow Q_0 \cup \{\#\}$

(1) m erreichbare Markierung

(2) $\forall\ t\ (t \in T \wedge t^{---} \leq m \rightarrow J(t) \leq I_2(t))$

(3) $\forall\ t\ (t \in T \wedge t^{---} \not\leq m \rightarrow J(t) = \#)$

Intervall-Petri-Netze (IPN)



t[a,b] 0 <= eft<= lft

eft → earliest firing time

lft → latest firing time

Zustand im Erreichbarkeitsgraphen

Platzdeskription → m (Markierung)

Transitionsdeskription → J (Zeitvektor)

z(m,J)

z = ((1,0,3), (2, T, 0, T))

Intervall-Petri-Netze (IPN)

- **formale Methode mit Zeitbezug**
- **Korrespondenz in der semantischen Deskription**
- **Überprüfung von Zeitanforderungen**
  - 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 (deadlock) nicht erreicht werden kann

$\sigma \rightarrow$ Transitionssequenz

Automatische Überführung von MSC in IPN

MSC → Prozess-Algebra
Intervall-Petri-Netze (IPN) → Zeit-Petri-Netz-Kalkül (tPBC)

- Beschreibung des MSC(IPN) mit Mitteln der Algebra
  - Grundelemente
  - Equivalenz-Relationen
  - Operationen

Toolunterstützung

- Unterstützung von modularer Strukturierung
- Vereinfachung von Hierarchiebildung
- Vereinfachung von Analysetechniken
- Verwendung von Designmustern

MSC → tPBC → IPN

Beschreibung von BMSC und HMSC durch tPBC

Zeit-Petri-Netz-Kalkül (tPBC)

Intervall Petri Box B ist eine Äquivalenzklasse B = [Σ]

$\Sigma = (P, T, F, V, m_0, I, \lambda)$

$\lambda : P \rightarrow \{ \{e\}, 0, \{x\} \}$ $\rightarrow$ entry,interne,exit Plätze

$\bullet\Sigma \{p \in P | \lambda(p) = \{e\}\}$ $\rightarrow$ Entry-Plätze

$\Sigma\bullet \{p \in P | \lambda(p) = \{x\}\}$ $\rightarrow$ Exit-Plätze

$\lambda(t) = 0$ $\rightarrow$ interne Transitionen

1. $\forall t \in T : \bullet t \neq 0 \neq t\bullet$ $\rightarrow$ jede t besitzt Vor- und Nachplatz
2. $\bullet\Sigma \neq 0$ $\rightarrow$ mindestens ein Entry-Platz
3. $\Sigma\bullet \neq 0$ $\rightarrow$ mindestens ein Exit-Platz
4. $\forall p \in \bullet\Sigma \; \forall t \in T : W(p,t) = 0$ $\rightarrow$ keine Kanten zu Entry-Plätzen
5. $\forall p \in \Sigma\bullet \; \forall t \in T : W(p,t) = 0$ $\rightarrow$ keine Kanten aus Exit-Plätzen
6. $\forall t \in T : (\lambda(t) \in W \Rightarrow \forall p \in P : W(p,t) = 1 \wedge W(t,p) = 1)$
   $\rightarrow$ nur einfache Gewichtsfunktion an Hierarchietransitionen
7. $\forall t \in T : (\lambda(t) \in W \Rightarrow \forall p \in P : W(p,t) = 0 \vee W(t,p) = 0)$
   $\rightarrow$ keine Seitenbedingungen bei Hierarchietransitionen

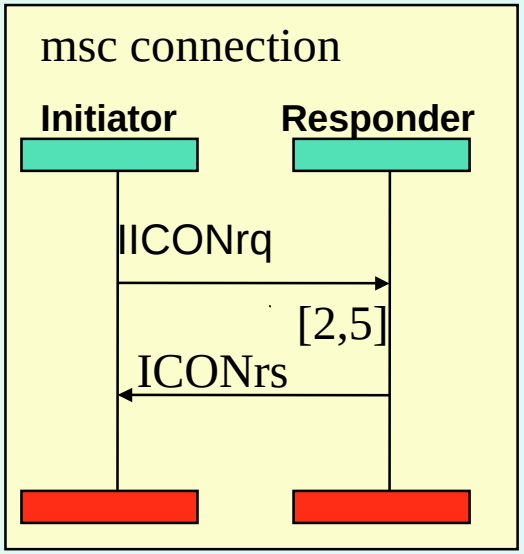
Zeit-Petri-Netz-Kalkül (tPBC) - Semantik

E::={ Basic Boxes }	{},{ß}	Aktion
	{ß}@[eft,lft]	Aktion mit Zeitintervall
	X	Variable

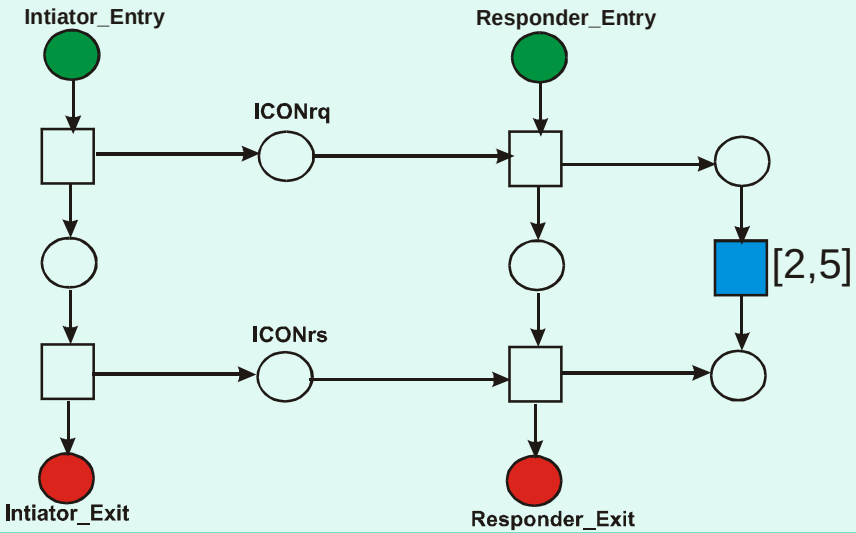
Basis-Operatoren

sequenzielle Konstrukte		
	Sequenz	E1 ; E2
	Choice	E1 [] E2
nebenläufige Konstrukte		
	Parallel	E1 || E2
	Synchronisation	E1 sy < Tiefe > Multimenge
Abstraktionskonstrukte		
	Restriktion	E1 rs Multimenge
	Scoping	[E1 : Multimenge]
	Relabelling	E [f]
hierarchische Konstrukte		
	Iteration	[<Tiefe> E1 * E2 * E3]
	Verfeinerung	E[X ← E]
asynchrone Konstrukte		
	Tie E1	E1 tie a

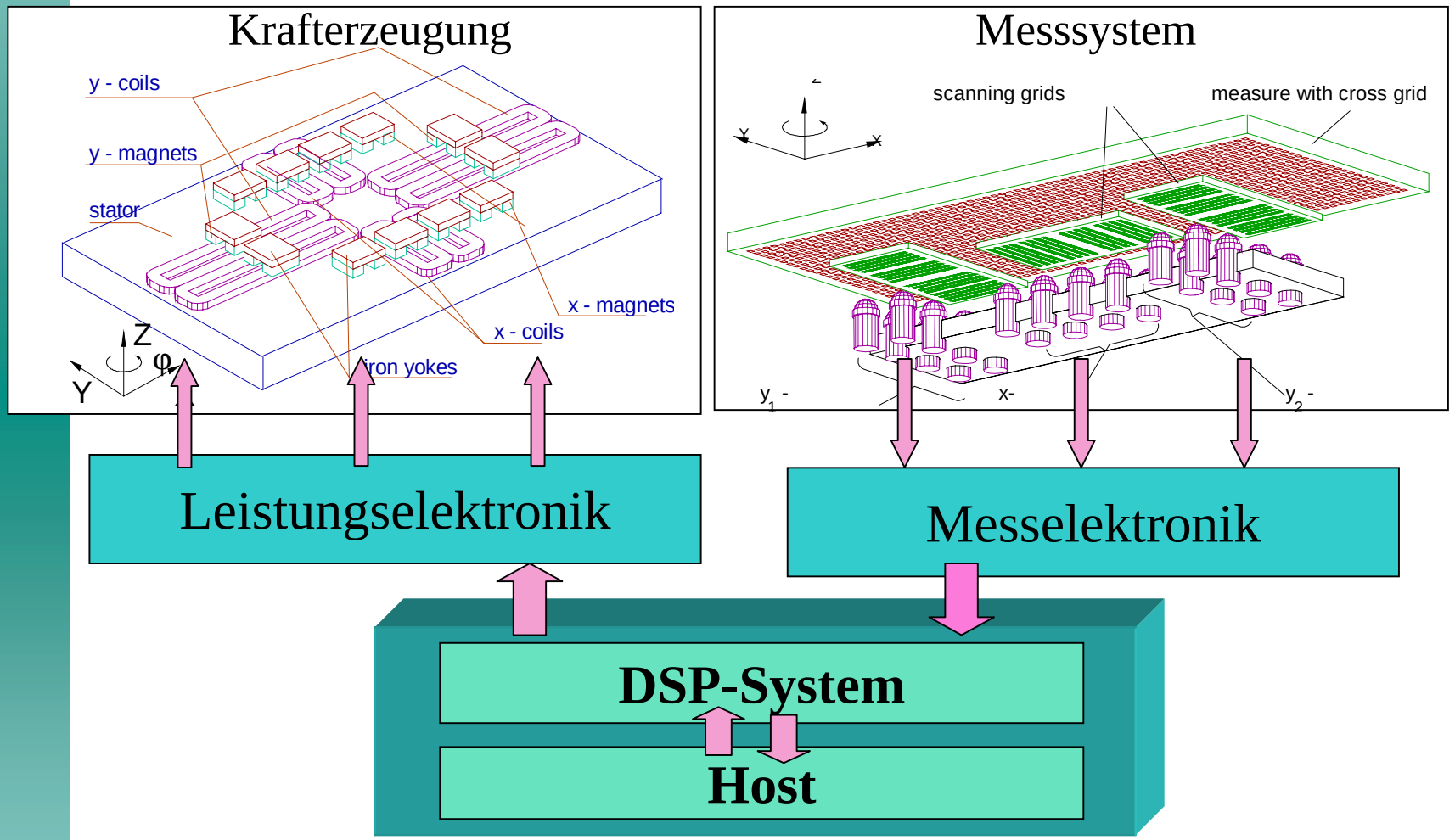
MSC → tPBC → IPN



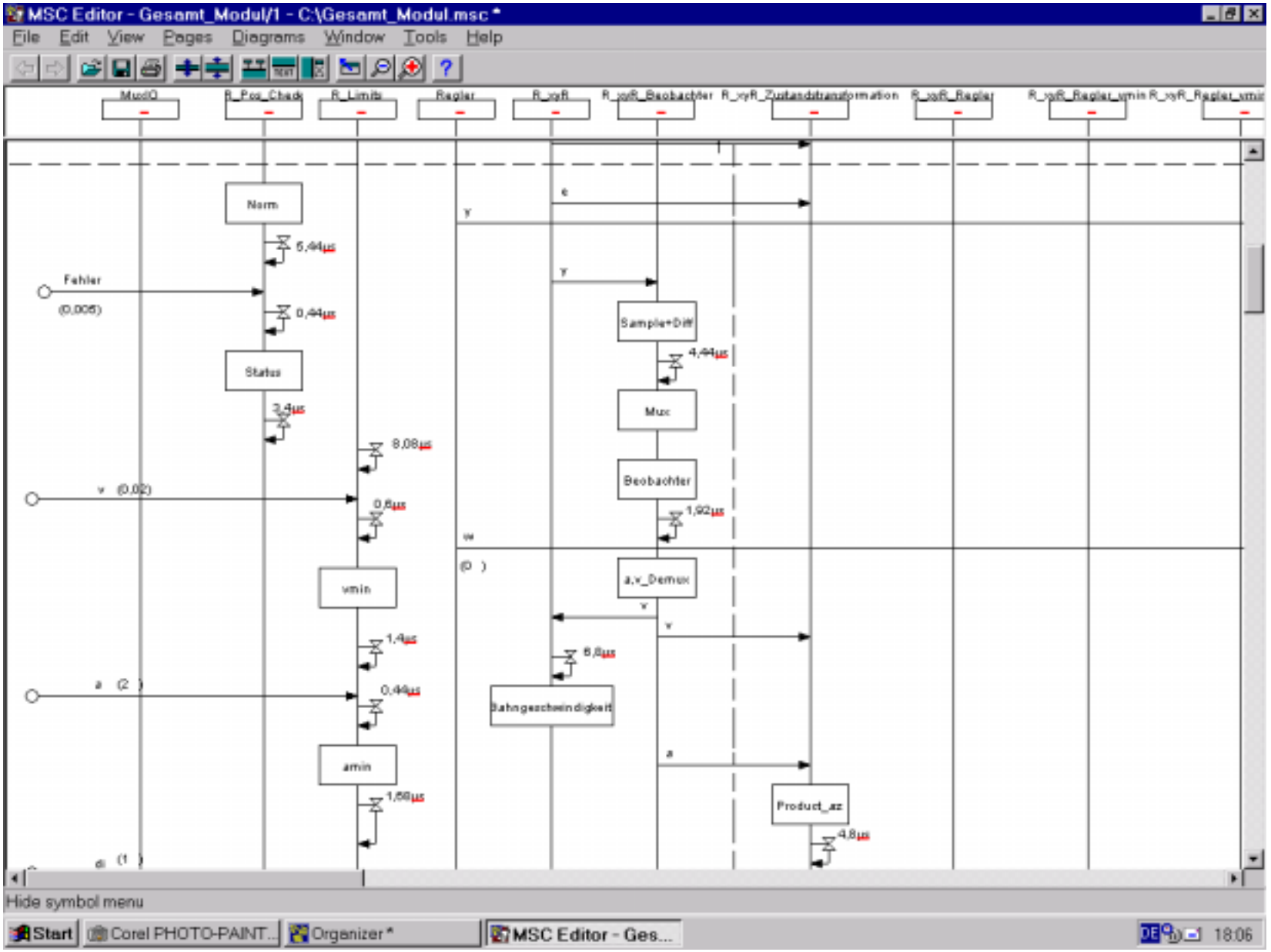

```
Initiator || Responder
[Initiator←(
    {ICONrq+}
    {ICONrs-}
)]
[Responder←
[{begin_int1,end_int1}:(
     {ICONrq-,begin_int1};
    {ICONrs+,end_int1}) ||
  (^begin_int1;{}@[2,5];^end_int1)]
]
```



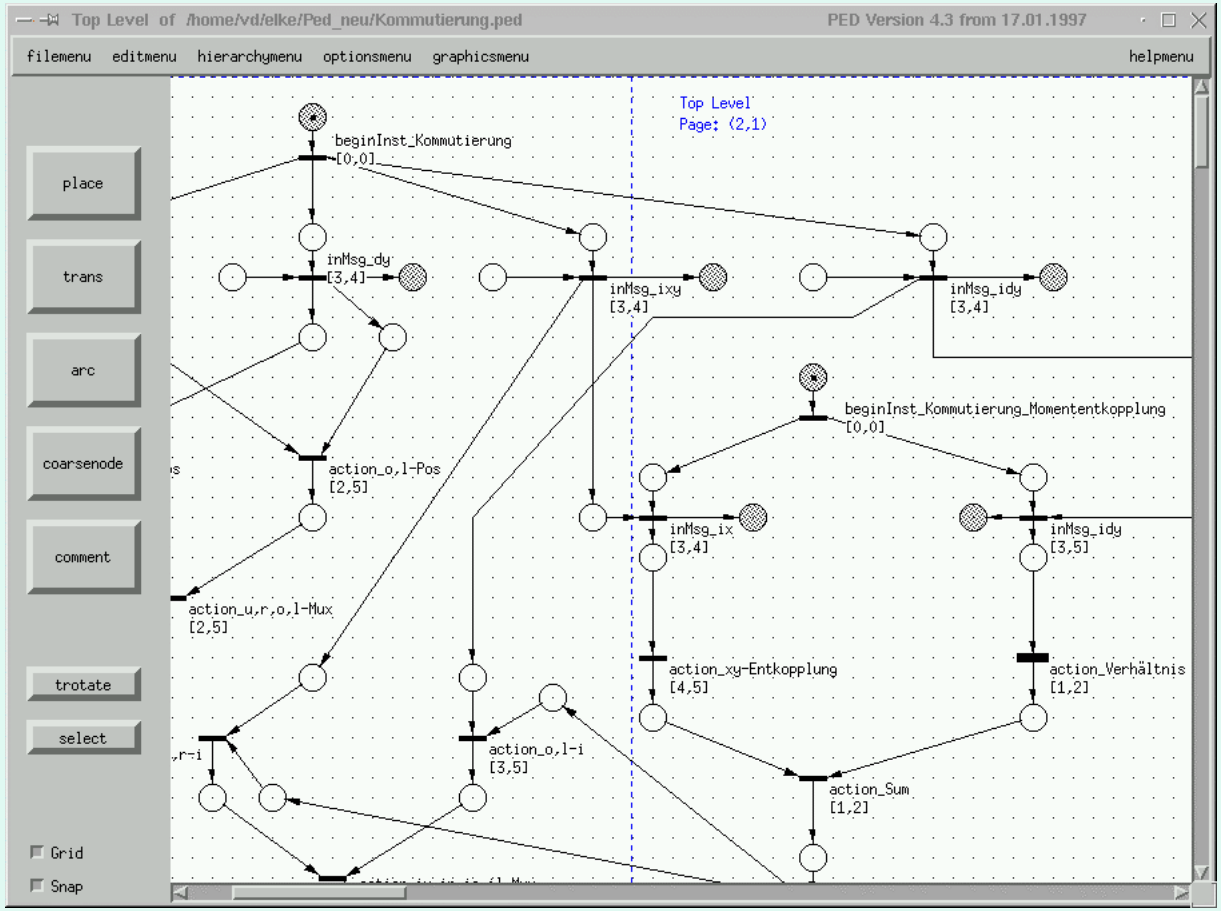
Anwendungsfeld → Mehrkoordinatenantriebe



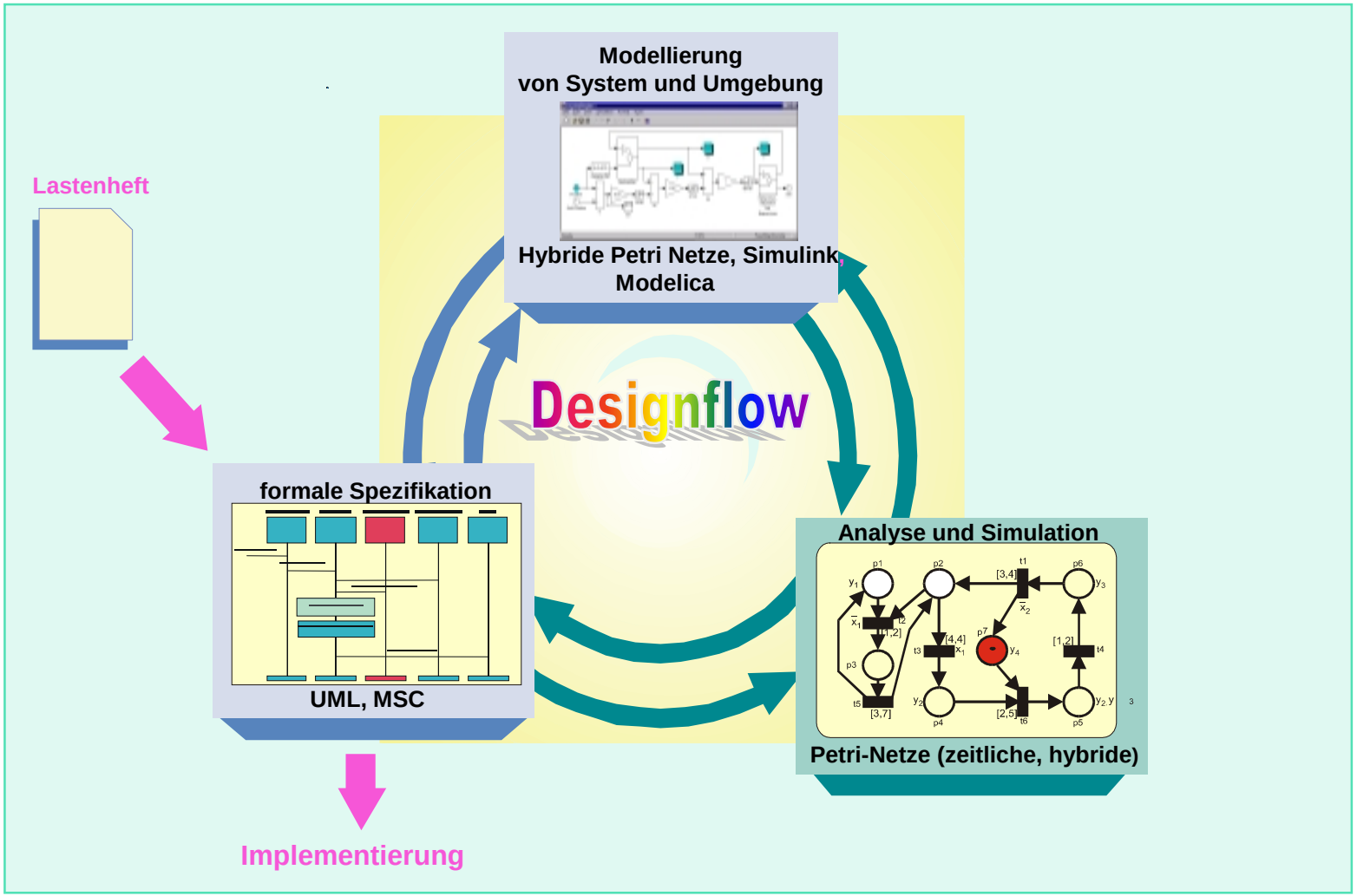
Mehrkoordinatenantrieb – MSC (TelelogicTau 4.0)



Mehrkoordinatenantrieb – IPN (PetriNetz Editor –PED)

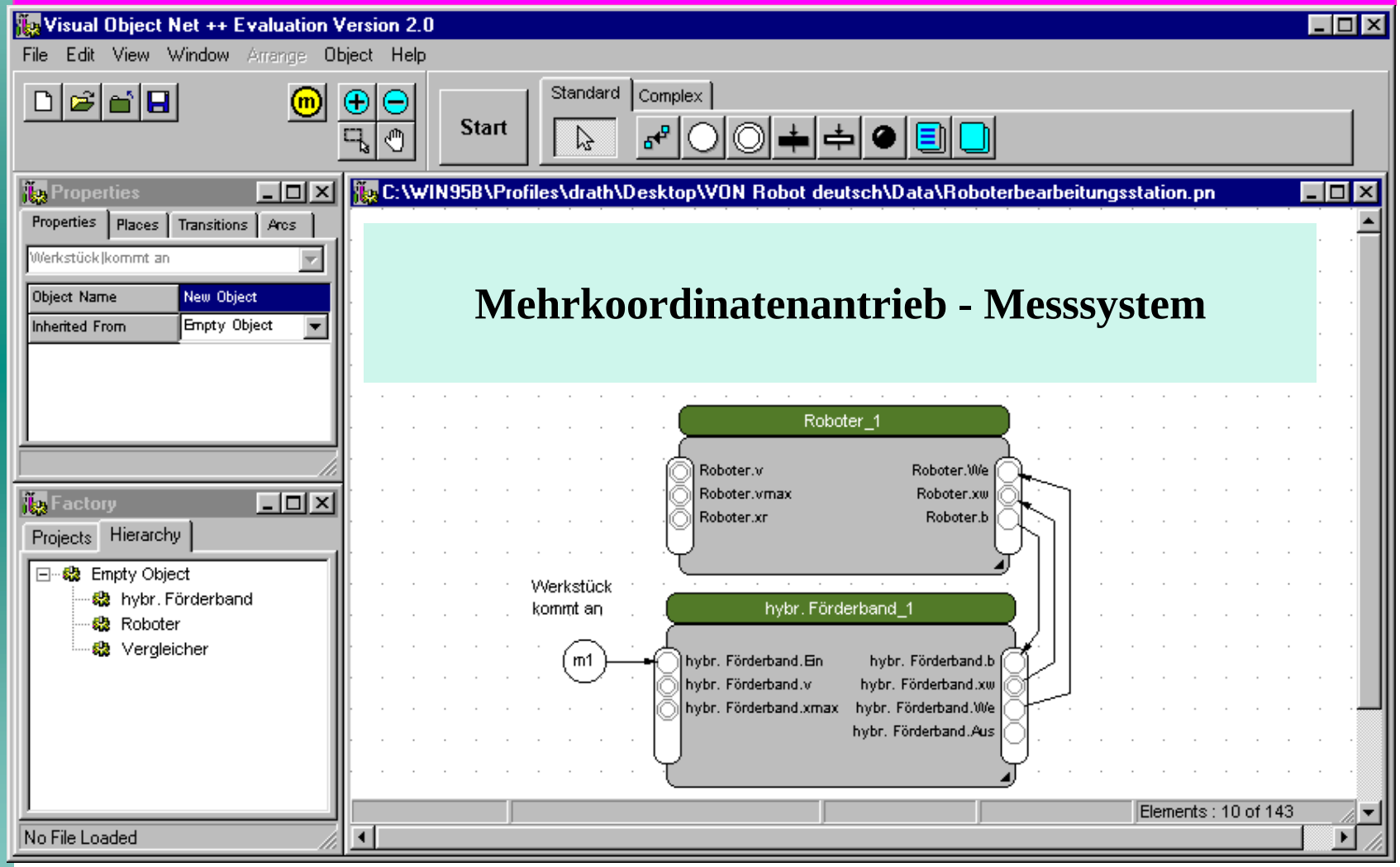


Entwurfsprozess hybrider Systeme



Modellierung und Simulation des Mehrkoordinatenantriebs - Hybride Petri Netze

(Visual Object Net ++)



Ausblick

Durchgängigkeit des Entwurfprozesses

- Integration unterschiedlicher Beschreibungsmechanismen
- Automatischer Wechsel zwischen den Kontexten
  - **MSC ➡ HPN**
  - **IPN ➡ MSC**

Formale Analyse

- Abbildung HPN ➡ IPN
  - Zustandsraumreduktion
  - Äquivalenz bezüglich ausgewählter Systemeigenschaften
    - Zeit
    - (Datenraten)