

Entwicklung einer Modellbibliothek für die Interaktion von Robotern in der MATLAB/DEVS Umgebung

Birger Freymann

Februar 2014

HOCHSCHULE WISMAR, UNIVERSITY OF APPLIED SCIENCES TECHNOLOGIES, BUSINESS AND DESIGN

b.freymann@stud.hs-wismar.de



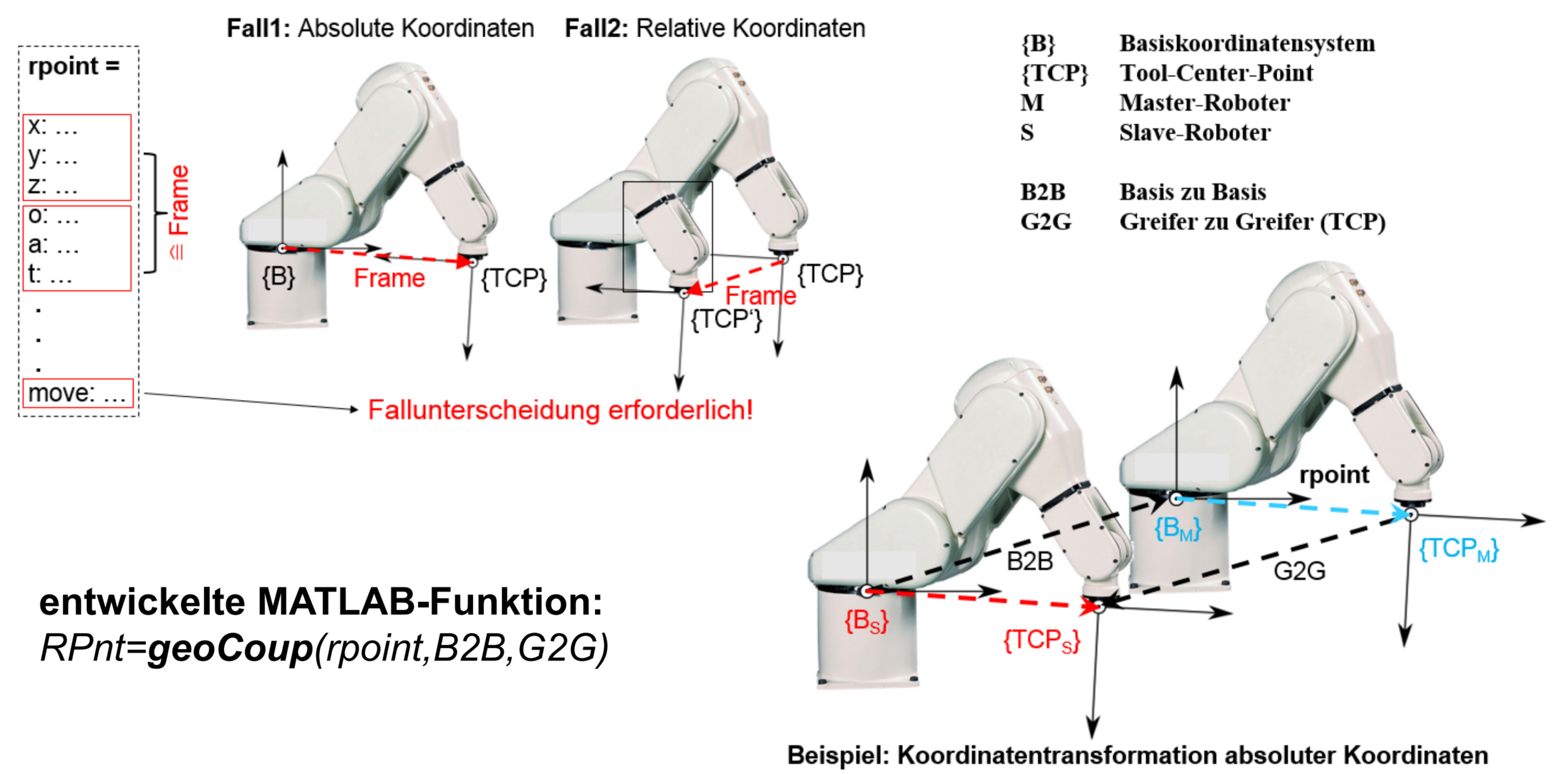
Kurz gesagt ...

Industrieroboter finden in unterschiedlichen Bereichen, von der Automobilindustrie bis hin zur Medizintechnik, Anwendung. Auf dem Markt sind unterschiedliche Hersteller vertreten, die individuelle Roboter und Software zur Programmierung der Roboter anbieten. Zumeist sind die Softwarelösungen herstellerspezifisch und lassen sich nicht für herstellerfremde Roboter verwenden. Langjährige Normierungsbemühungen für die explizite Roboterprogrammierung, wie die IRL (Industrial Robot Language) und deren Nachfolger die PLR (Programming Language for Robots), wurden durch die Roboterhersteller bis heute zumeist nicht umgesetzt. Dies führt zu einer Einschränkung bei einer gemeinsamen Nutzung von Robotern unterschiedlicher Hersteller. Aufbauend auf dem Simulation Based Control (SBC) Framework für die ereignisorientierte Steuerungsentwicklung wird ein Ansatz zur Lösung dieses Problems vorgestellt. Der SBC-Ansatz baut auf dem Rapid Control Prototyping (RCP) auf und unterstützt eine schrittweise und durchgängige modellbasierte Steuerungsentwicklung von der frühen Planungsphase bis zum operativen Betrieb. Die Steuerungsentwicklung erfolgt unabhängig von konkreten Robotertypen. Die Integration erfolgt über eine spezielle Schnittstelle, welche einmalig für den jeweiligen Robotertyp zu entwickeln ist. Die ereignisorientierte Steuerungsentwicklung erfolgte in diesem Fall mit dem Discrete Event System Specification (DEVS)

Formalismus und einer darauf basierenden MATLAB-DEVS-Toolbox. Diese unterstützt eine systemtheoretischbasierende Modellierung und Simulation ereignisorientierter Systeme. Hinsichtlich eines durchgängigen Einsatzes der Modelle bis zum operativen Betrieb unter Echtzeitbedingungen wurde der DEVS konforme PDEVS-RCP-Formalismus definiert. Des Weiteren wurden unterschiedliche Interaktionsprinzipien für Industrieroboter analysiert und aufgrund ihrer praktischen Relevanz die geometrische Kopplung als zu untersuchendes Interaktionsprinzip ausgewählt. Ein typischer Anwendungsfall der geometrischen Kopplung ist das Lastteilungsverfahren (load sharing). Hierbei wird ein schweres Bauteil von mehreren Robotern bewegt. Dies spart im Vergleich zu einem größeren Roboter, der die gleiche Last bewegen könnte, Kosten und Energie. Gleichmaßen ergeben sich aber auch neue Herausforderungen, wie z.B. die Bewegungssynchronisation der geometrisch gekoppelten Roboter. Zur Entwicklung der Modellbibliothek wurde die SBC-Struktur einer allgemeinen Robotersteuerung analysiert. Wiederverwendbare Komponenten wurden identifiziert und in eine Modellbibliothek integriert. Ausgehend von einem Anwendungsbeispiel wurden bestehende Grenzen der entwickelten Modellbibliothek analysiert und weitergehende Lösungsansätze aufgezeigt.

DEVS-Modellbibliothek & Anwendungsbeispiel

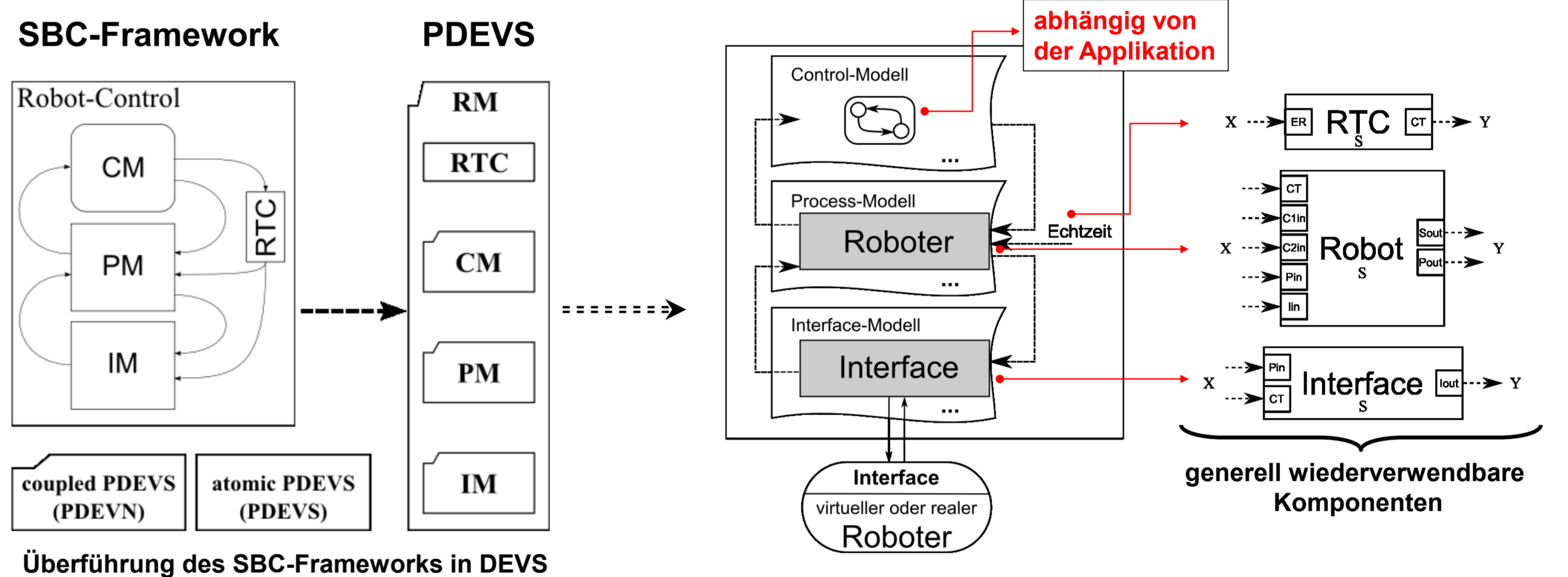
Koordinatentransformationen zur geometrischen Kopplung von Robotern



entwickelte MATLAB-Funktion:
 $RPnt=geoCoup(rpnt, B2B, G2G)$

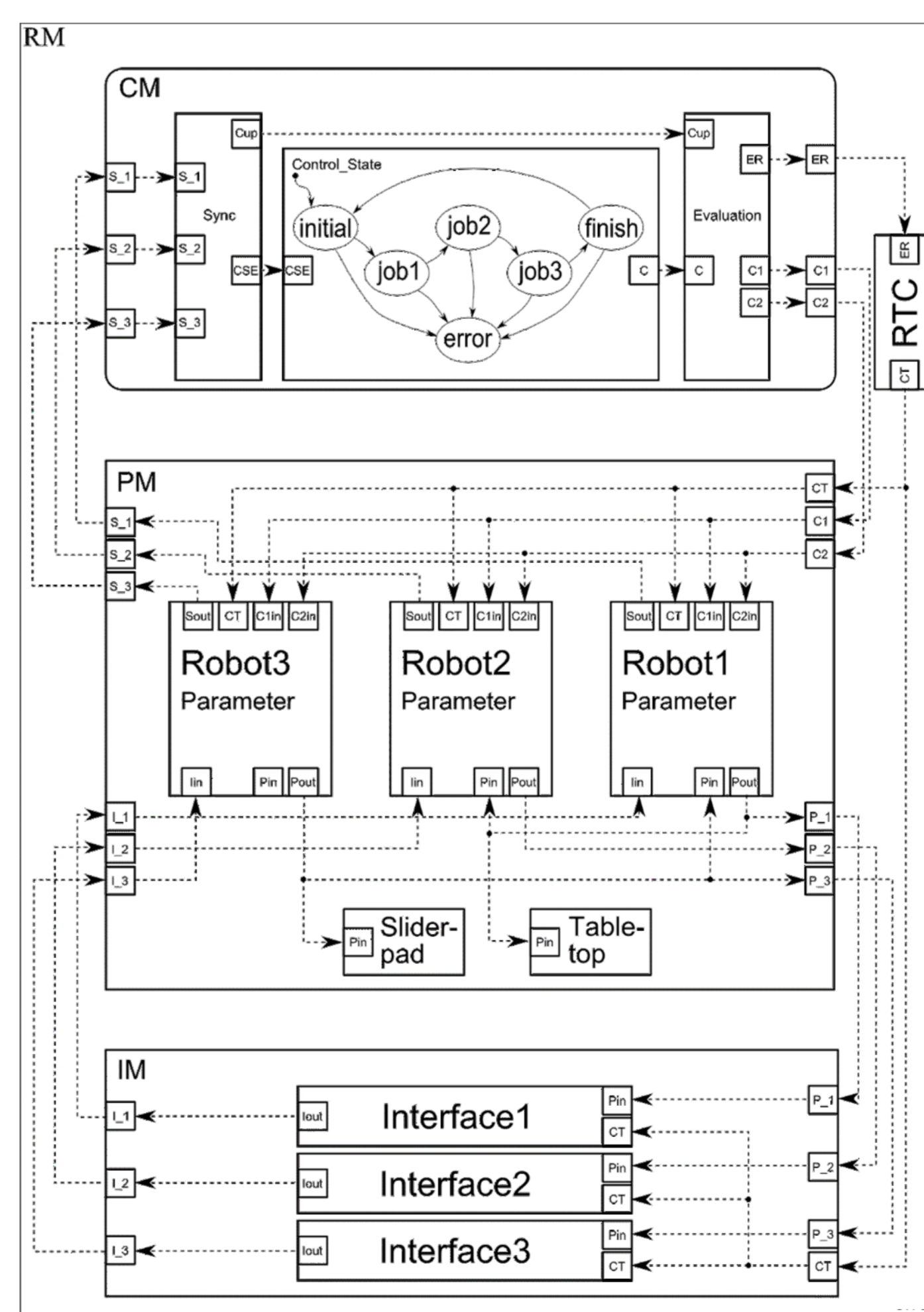
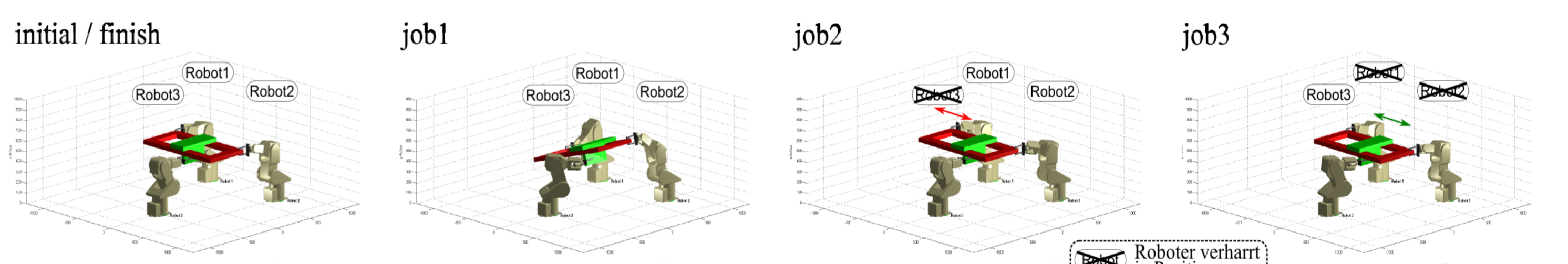
Beispiel: Koordinatentransformation absoluter Koordinaten

SBC-Grundstruktur einer Robotersteuerung & Analyse auf wiederverwendbare Komponenten



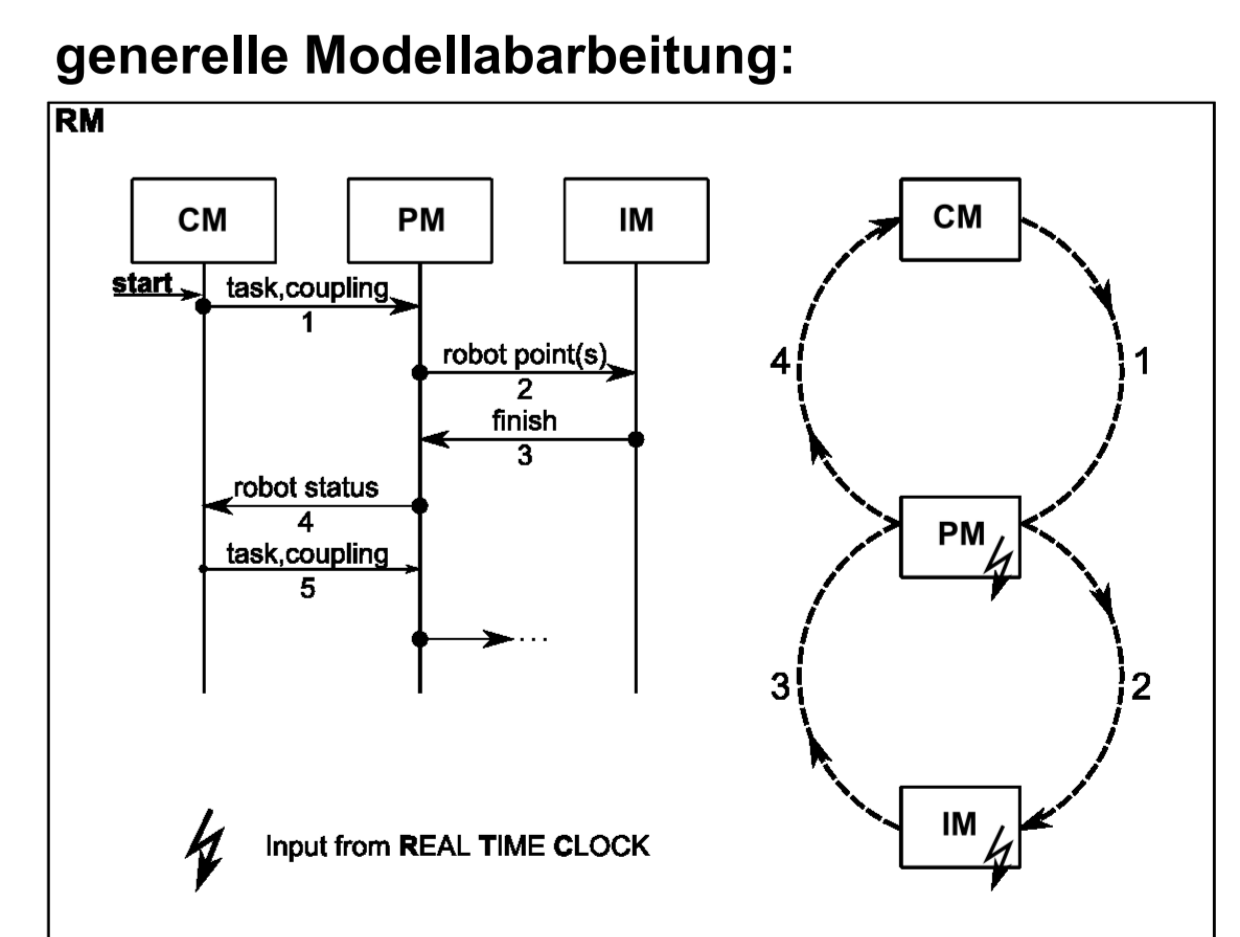
Überführung des SBC-Frameworks in DEVS

Anwendungsbeispiel: Interaktion von 3 Robotern mit geometrischer Kopplung

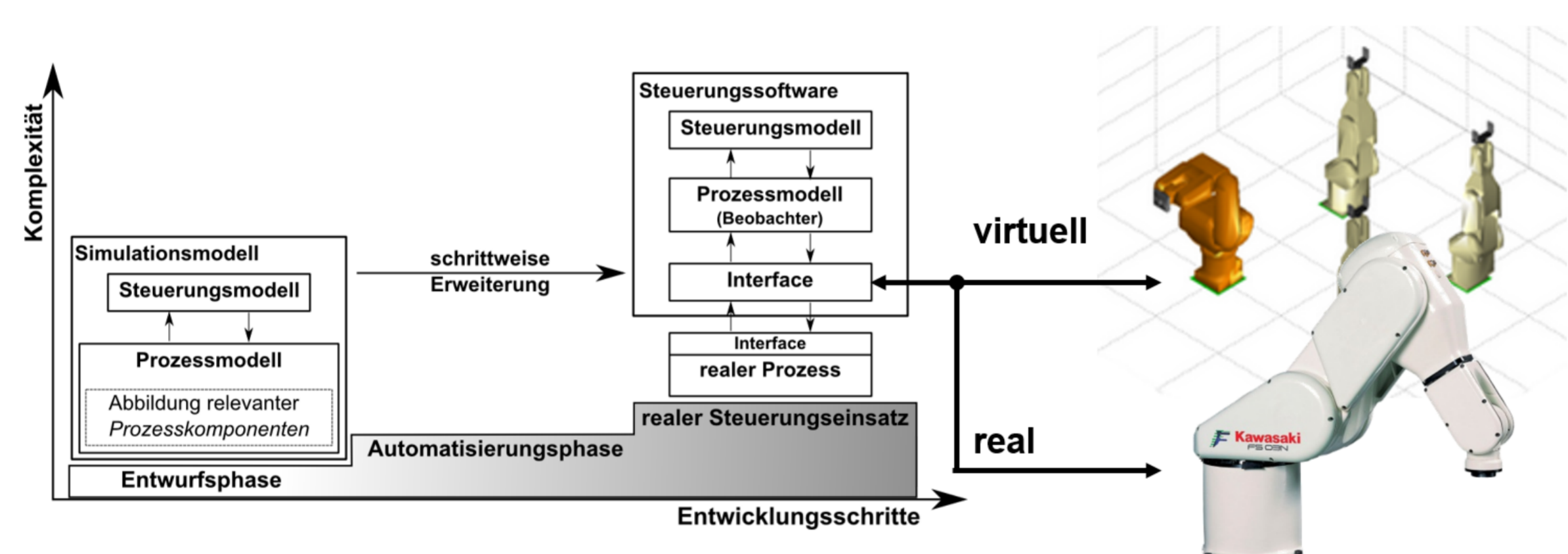


Master-Slave-Struktur					
TASKS	Robot	1	2	Kopplung Task zeigt ...	
initial	M	M	M	nein	individuelle Bewegung
job1	M	S(3)	S(1)	ja	positionsab- und unabhängige geo. K.
job2	-	M	S(1)	ja	koordinierte Bewegung der Slaves
job3	M	-	-	nein	ausschließlich Master
finish	M	M	M	nein	individuelle Bewegung

Legende			
M	Master	CM	Control - Modell
S(X)	Slave von Roboter X	PM	Process - Modell
-	verharrt in Position	IM	Interface - Modell



Simulation Based Control (SBC) Framework



- durchgängige Nutzung von Simulationsmodellen während der gesamten Steuerungsentwicklung
- aufbauend auf dem Rapid Control Prototyping (RCP)
- simulatives Testen vom Entwurf bis zum Betrieb

PDEVS-RCP-Spezifikation

$PDEVS - RCP = (X, Y, S, \delta_{int}, \delta_{ext}, \delta_{con}, \lambda, ta, A)$

X $X = X_{model} \cup X_{clock}$
 $X_{model} = \{(p, v) | p \in IPorts, v \in X_p\}$
 $X_{clock} = \{("clock", v) | v \in R^+\}$
 Menge aller Eingangswerte
 Menge der Eingangsereignisse des Modells
 Menge der Eingangsereignisse von einer Echtzeituhr

Y Menge der Ausgangswerte

S Menge der Zustandswerte

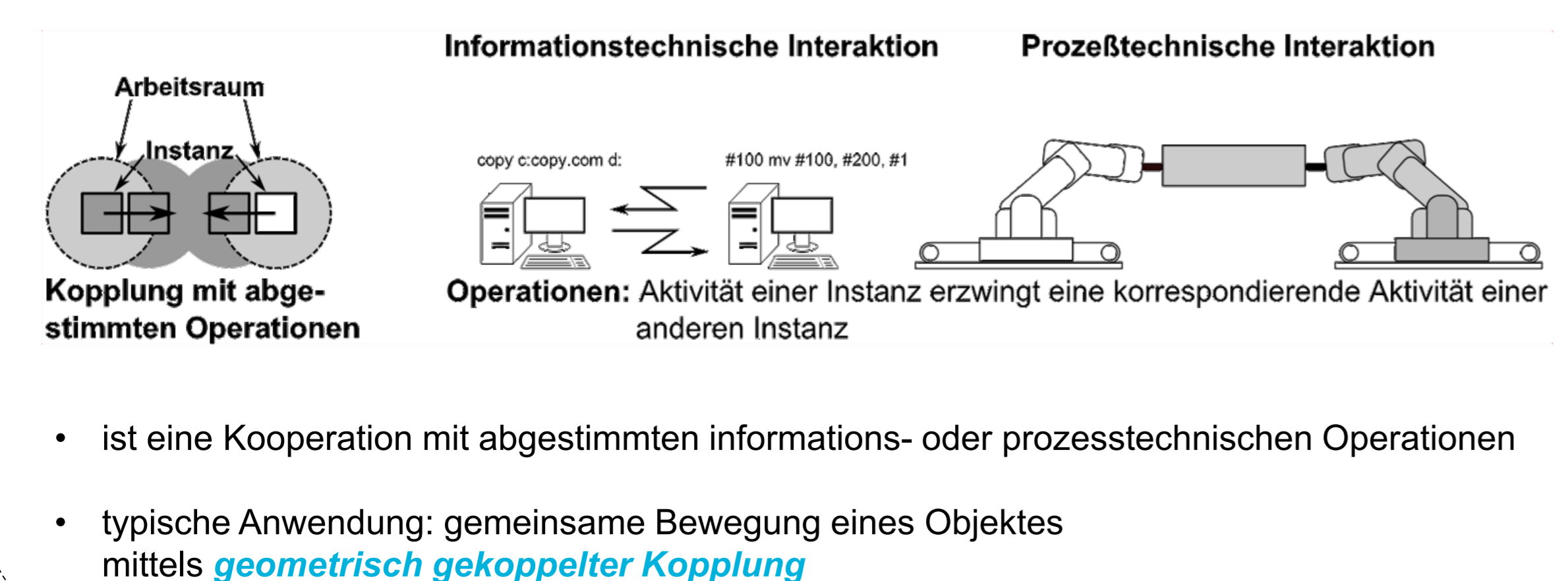
$\delta_{int} \delta_{int} : S \rightarrow S$
 $\delta_{ext} \delta_{ext} : Q \times X \rightarrow S$
 $Q = \{(s, e) | s \in S, 0 \leq e \leq ta(s)\}$
 e (verstrichene Zeit seit letztem Ereignis)
 $\delta_{con} \delta_{con} : Q \times X \rightarrow S$
 interne Zustandsüberföhrungsfunktion
 externe Zustandsüberföhrungsfunktion
 total state
 time elapsed
 Zustandsüberföhrungsfunktion bei zeitgleichen Ereignissen (confluent transition function)

$\lambda \lambda : S \rightarrow Y \times A$
 kombinierte Ausgabe- und Aktivitätsfunktion

$ta ta : S \rightarrow R^+_{0, \infty}$
 Zeitfortschrittsfunktion

A Menge der ausführbaren Aktivitäten

Interaktion: Kopplung mit abgest. Operationen



Zusammenfassung

- SBC-Framework unterstützt Entwicklung ereignisorientierter, interagierender Robotersteuerungen mit DEVS
- Allgemeine Modellbibliothek für geometrisch gekoppelte Roboter umgesetzt und am Beispiel getestet

Noch nicht vollständig gelöst:

- Probleme der zeitlichen Synchronisation
- feste Kopplungen (feste Master-Slave-Beziehungen)

AUSBLICK: Dynamische Kopplungen:

- Realisierung mit SES + DEVS oder
- strukturvariablem DEVS

Link zur Master-Thesis: http://www.mb.hs-wismar.de/~pawel/Study_DiplomBelege/

- Betreuer: Prof. Dr.-Ing. Thorsten Pawletta, FIW, RG CEA
- Betreuer: M.Eng. Tobias Schwatinski, FIW, RG CEA