

Übersetzung von UML-Software-Spezifikationen in Simulationsmodelle

Stefan Walter

swalter@dspace.de

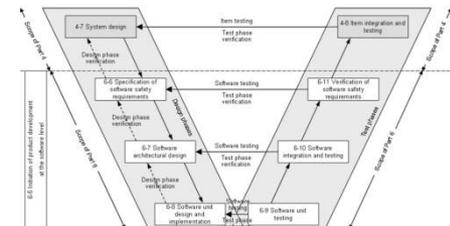
Lehrstuhl für Informationstechnik, insb. Realzeitsysteme
FernUniversität in Hagen



Fachtagung Echtzeit 2014 in Boppard am Rhein



- **Immer komplexer werdende Anforderungen an Softwaresysteme**
 - modellgetriebene Softwareentwicklung gewinnt zunehmend an Bedeutung.
- **Unified Modeling Language (UML) zur Beschreibung von Softwarearchitekturen**
- **UML ist ein sehr mächtiges und generisches Instrument**
 - Modelle besitzen viel Interpretationsspielraum.
 - zur Modelltransformation nur bedingt geeignet
- **UML-Profile, um den Standardsprachumfang von UML zu erweitern und zu konkretisieren**
- **Sehr frühes Testen auf Basis der entwickelten Modelle**
 - z.B. quantitative und qualitative Analysen von Softwarearchitekturen
 - einfachere und auch kostengünstigere Behebung von Fehlern
- **Erstellen von Simulationsmodellen aus UML Softwarearchitekturen**





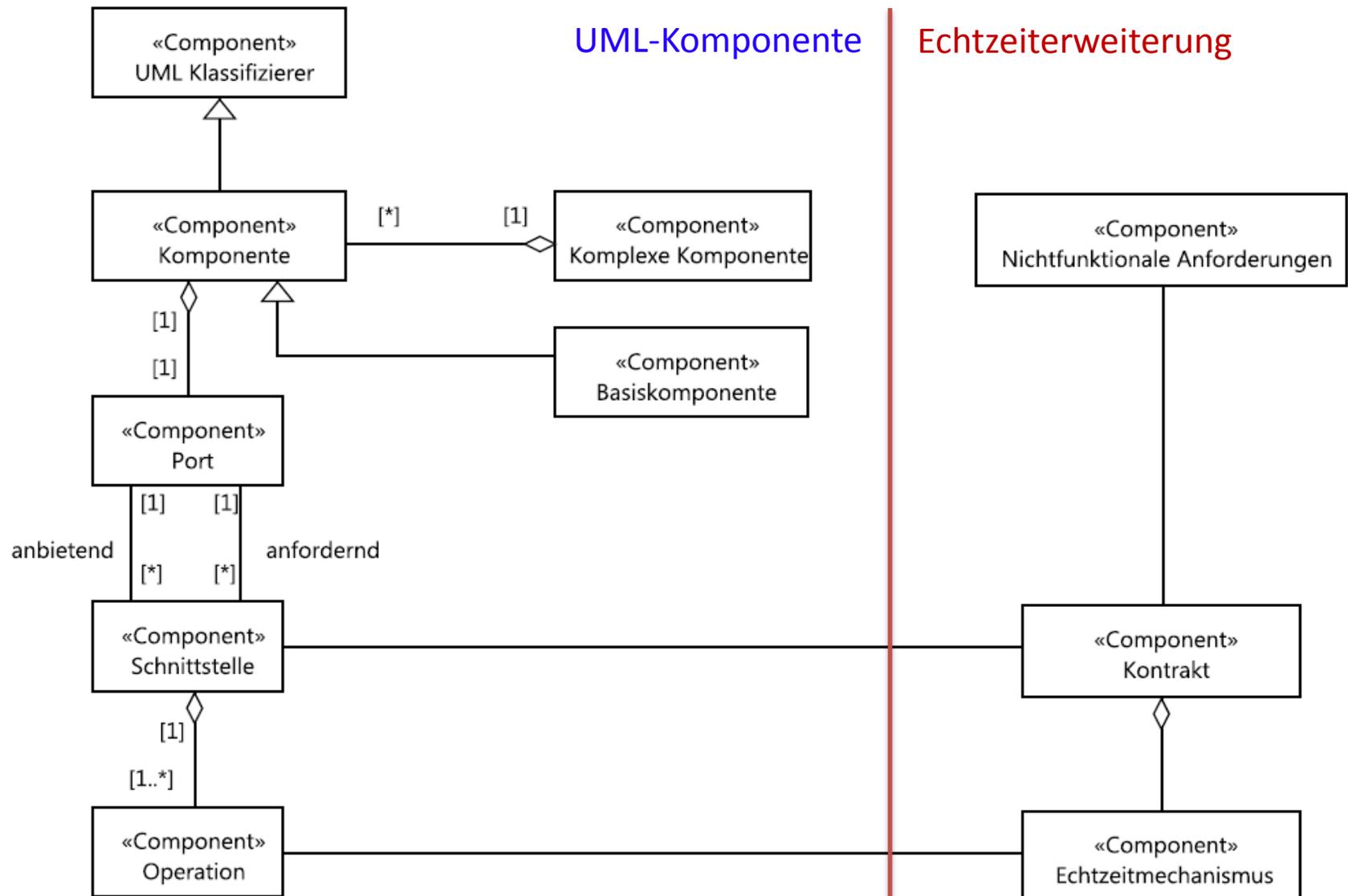
- **Masterarbeit am Lehrstuhl für Informationstechnik, insb. Realzeitsysteme der FernUniversität in Hagen**
 - Machbarkeitsuntersuchung der Übersetzung von Softwarearchitekturen von UML nach Simulink
 - Berücksichtigung von UML-Modellen eines Echtzeit-UML-Profiles (UML-RT-Profiles), entwickelt am Lehrstuhl für Informationstechnik der FernUniversität in Hagen
 - Beschränkung auf UML-Komponenten- und UML-Sequenzdiagramme, sowie deren echtzeitspezifische Erweiterung
- **Abgrenzung:**
 - Keine durchgängige Implementierung der Transformation
 - Beschränkung ausschließlich auf plattformunabhängige Softwaremodelle
 - Keine Berücksichtigung von Softwarearchitekturen für verteilte Systeme



- Einführung
- Übersetzen von UML-Komponentendiagrammen nach Simulink
- Übersetzen von UML-Sequenzdiagrammen nach Simulink
- Zusammenfassung



- Einführung
- **Übersetzen von UML-Komponentendiagrammen nach Simulink**
- Übersetzen von UML-Sequenzdiagrammen nach Simulink
- Zusammenfassung





- UML-Komponente

$$Komp = (N, P, S_{anf}, S_{anb}, V, O)$$



- Generische UML-Komponente

$$Komp = (N, \underline{P_{DataOut}}, \underline{P_{DataIn}}, \underline{P_{Func}}, V, O)$$

- Ports:

- *DataIn*

- Set[AttributName]([Variable] : [Typ])*

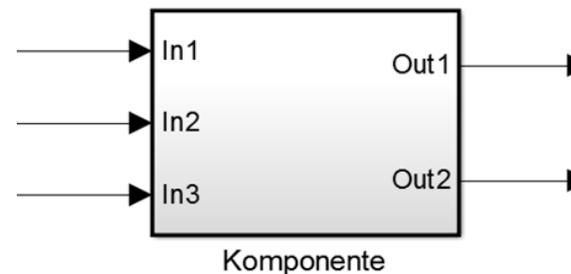
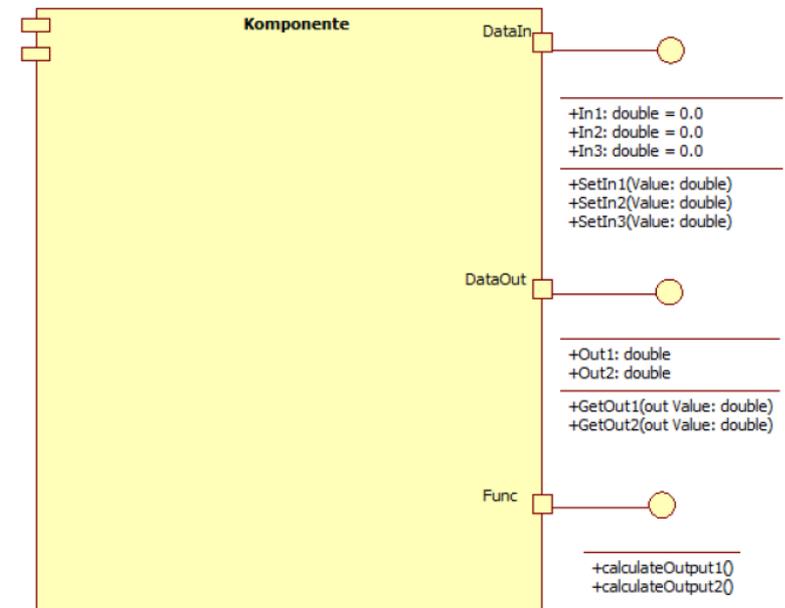
- *DataOut*

- Get[AttributName](out[Variable] : [Typ])*

- *Func*

- Atomares Simulink-Subsystem

$$SubSys = (N, P_{Ein}, P_{Aus}, Para)$$





- Periodische Komponenten:**

- Def: $S_i < F_i \leq D_i; D_i \leq P_i$

$$\underline{SubSys} = (N, P_{Ein}, P_{Aus}, Para)$$

S: Startzeitpunkt

F: Endzeitpunkt

D: Fertigstellungstermin

P: Periode

MaxRT: Maximale Antwortzeit

T_{min} : minimaler Abstand zwischen zwei Aktivierungen

- Aperiodische Komponenten:**

- Def: $S_{i,k} < S_{i,k+1} \implies S_{i,k} < MaxRT \leq S_{i,k+1}$

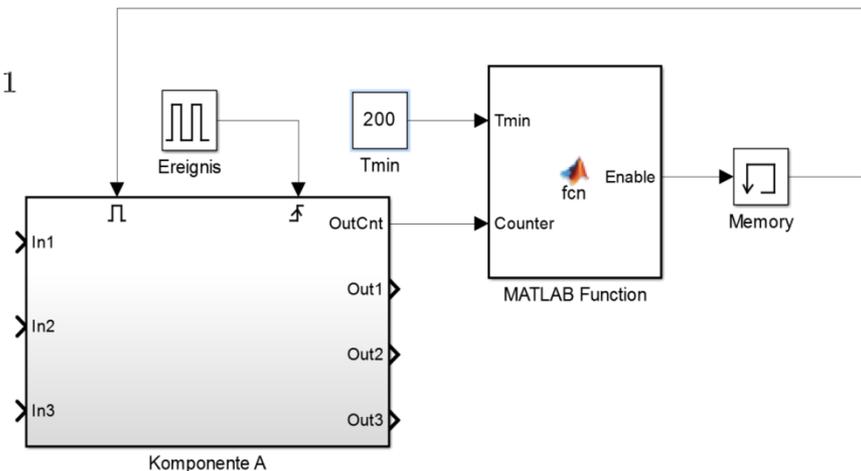
$$\underline{SubSys}_{Trg} = (N, P_{Ein}, P_{Aus}, P_{Trg}, Para)$$

- Sporadische Komponenten:**

- Def: $S_{i,k+1} > S_{i,k} + T_{min}, S_{i,k} < D < S_{i,k+1}$

- Aktivierbares, ereignisgesteuertes Simulink-Subsystem

$$\underline{SubSys}_{ActTrg} = (N, P_{Ein}, P_{Aus}, P_{Trg}, P_{Act}, Para)$$

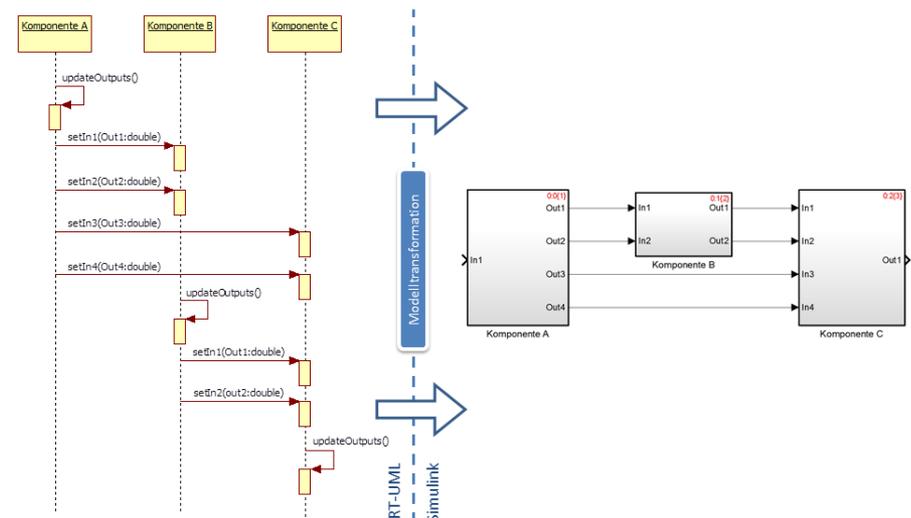


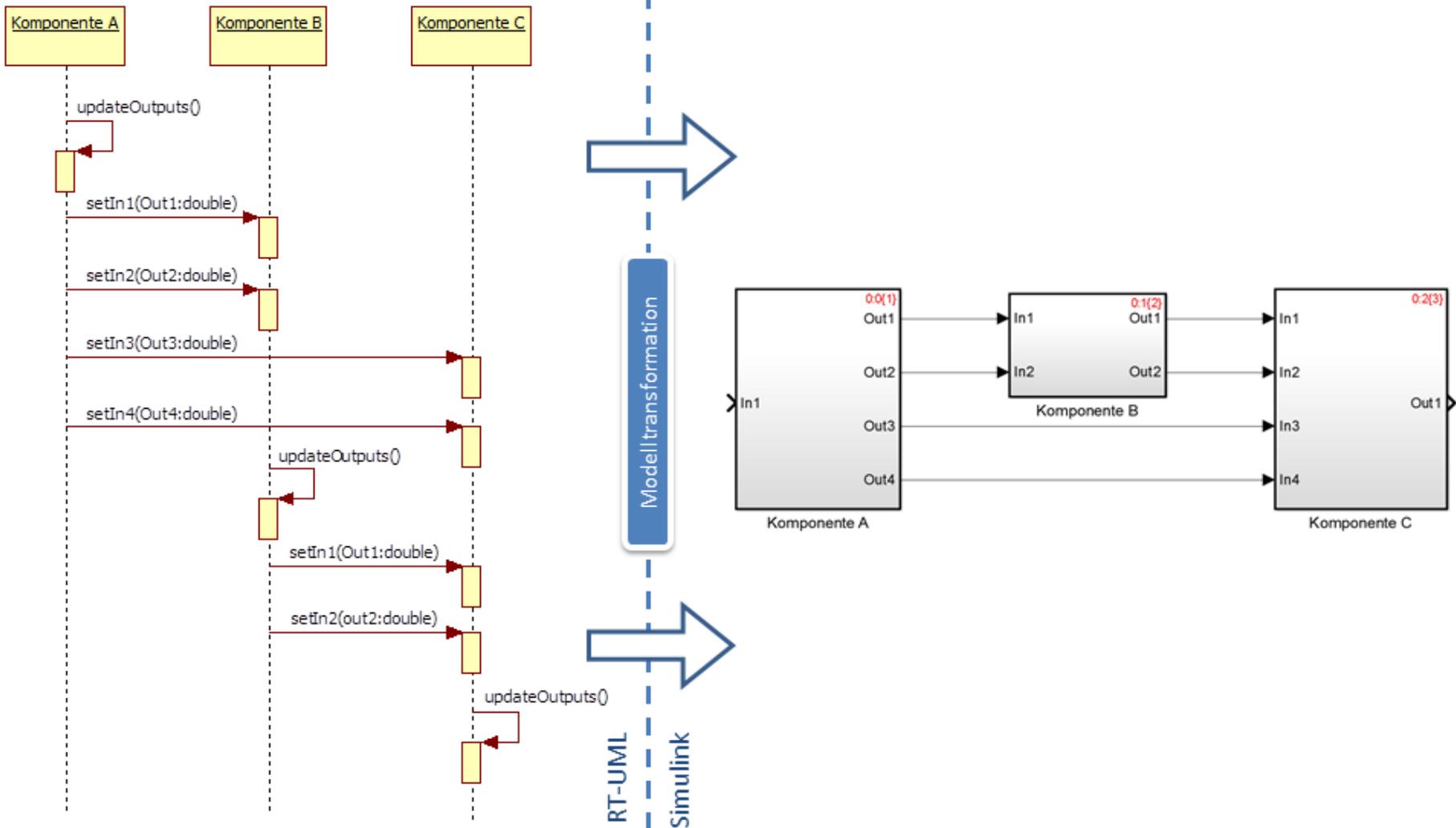


- Einführung
- Übersetzen von UML-Komponentendiagrammen nach Simulink
- **Übersetzen von UML-Sequenzdiagrammen nach Simulink**
- Zusammenfassung



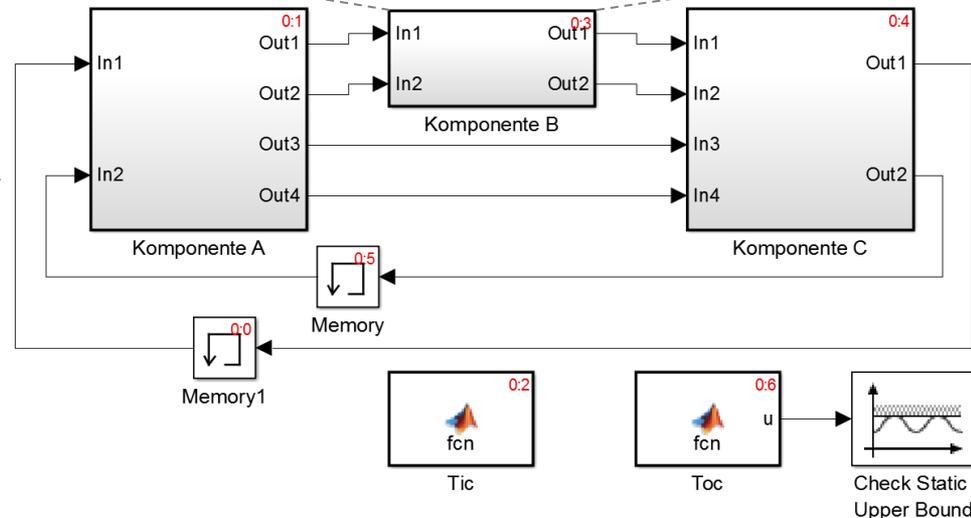
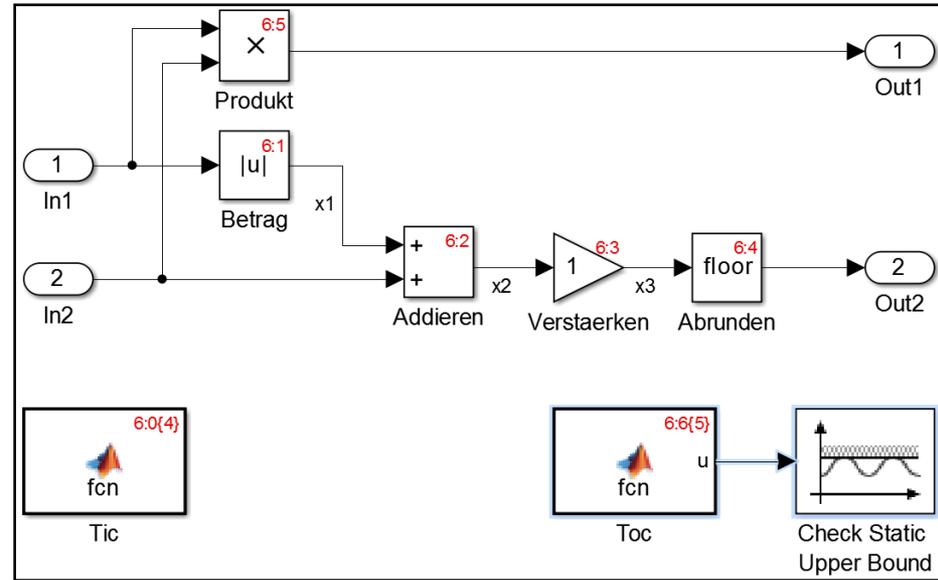
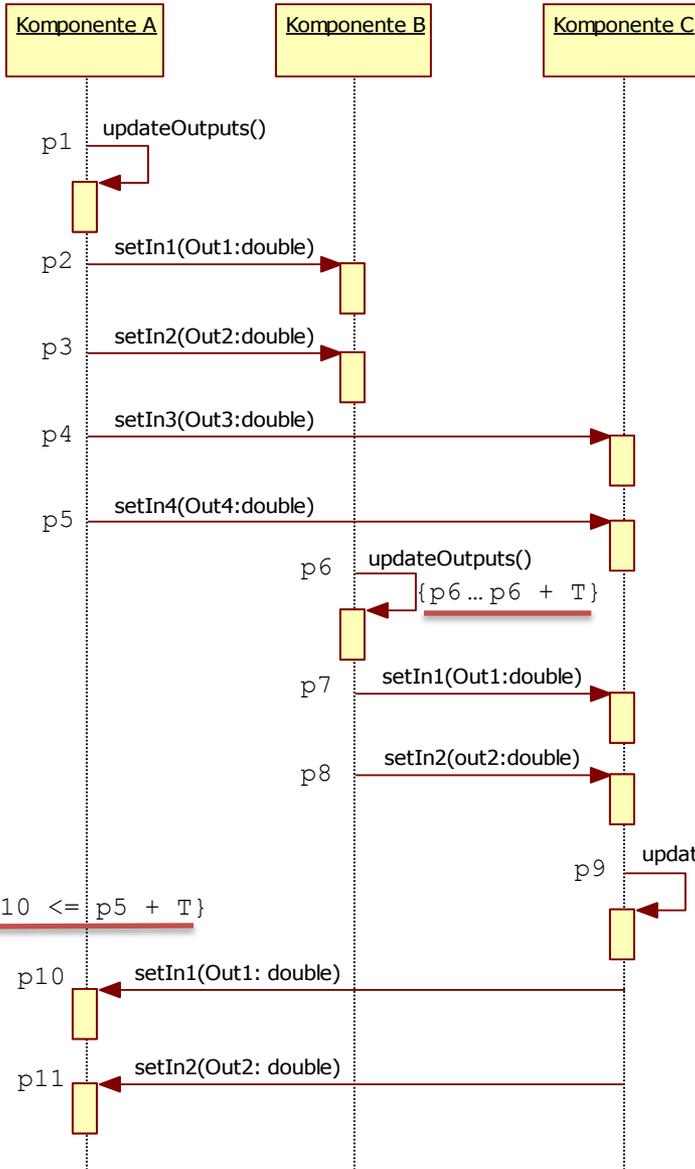
- **Ergänzung zum statischen Komponentendiagramm**
- **Festlegung der Ausführungsreihenfolge von Operationen**
- **Festlegung der Operationen einer Komponente, die im Kontext aufgerufen werden**
- **Zeitlicher Zusammenhang zwischen den Ausführungszeitpunkten von Operationen**
- **Kontrollfluss:**
 - Optionale/ Alternative Ausführung:
 - Iterative Ausführung
- **Echtzeitaspekte:**
 - Zeitbedingungen
 - Dateninvarianten
 - Vor- und Nachbedingung





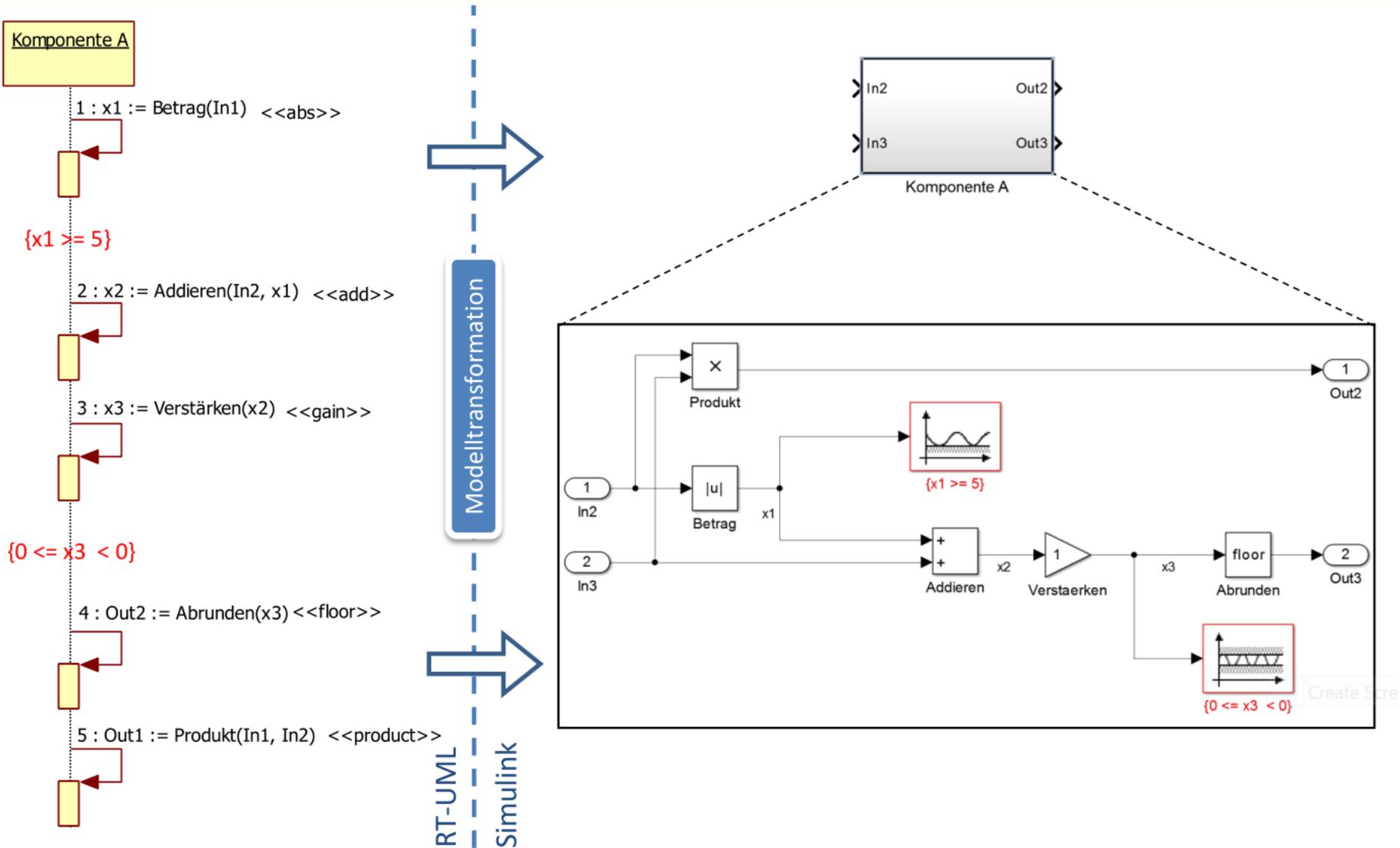


- **Zeitbedingungen:**
 - Definieren von Zeitmarkierungen im Sequenzdiagramm
 - Einschränken der Abhängigkeit zwischen Zeitmarkierungen
 - Überprüfung der Zeitmessung in Simulink mittels „Model Verification“-Blöcken
- **Dateninvarianten:**
 - Definition von Zwangsbedingungen zur Einschränkung des Wertebereichs von Variablen
 - Überprüfung nur an bestimmten Stellen möglich (keine permanente Überwachung)
 - Überprüfung in Simulink mittels „Model Verification“-Blöcken
- **Vor- und Nachbedingung:**
 - Zwangsbedingungen als Vor- und Nachbedingung von Operationen
 - Aufspalten in Simulink-Subsysteme für Vor-/ Nachbedingung und Operationen
 - Überprüfung in Simulink mittels „Model Verification“-Blöcken



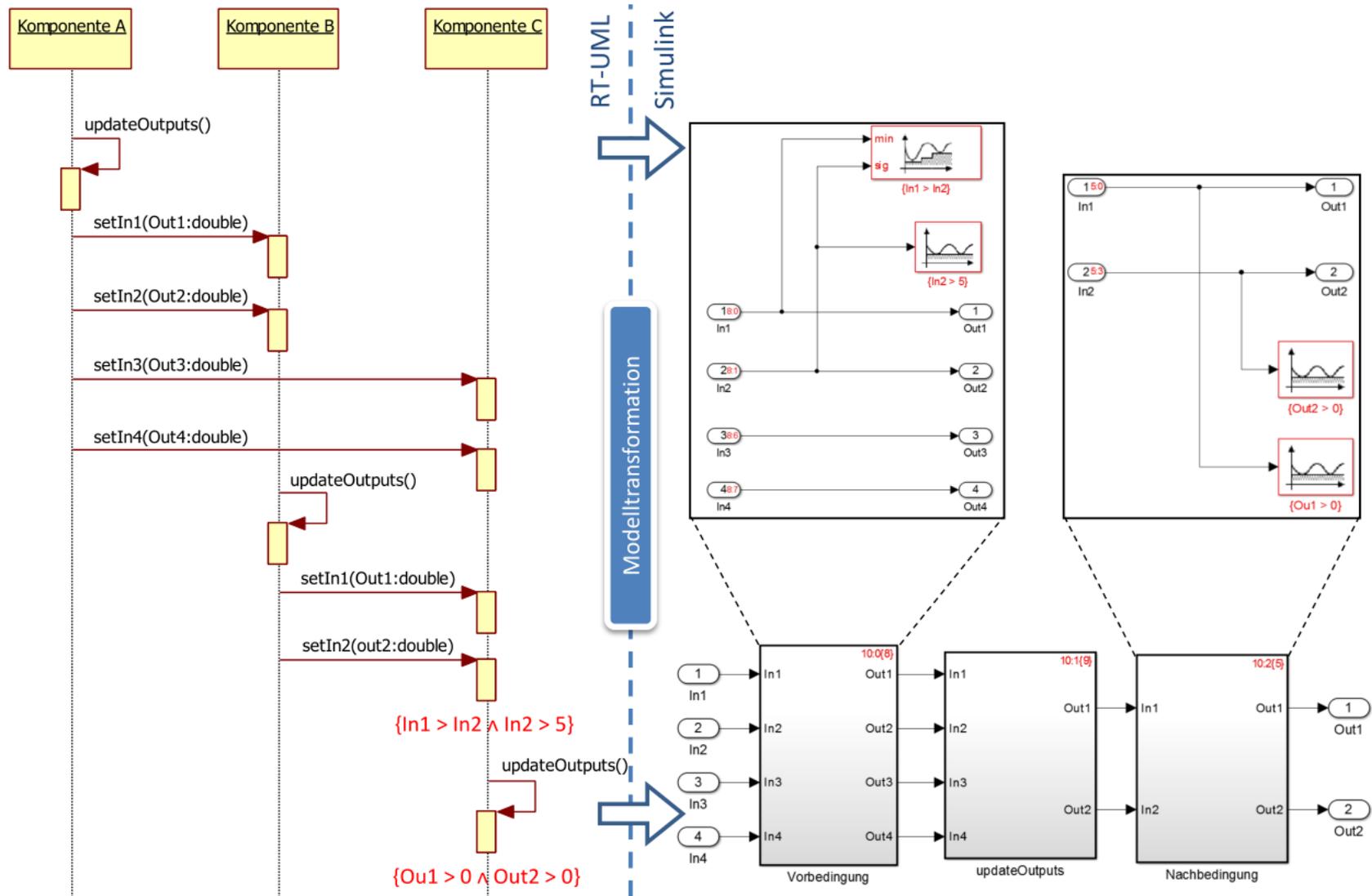


- **Zeitbedingungen:**
 - Definieren von Zeitmarkierungen im Sequenzdiagramm
 - Einschränken der Abhängigkeit zwischen Zeitmarkierungen
 - Überprüfung der Zeitmessung in Simulink mittels „Model Verification“-Blöcken
- **Dateninvarianten:**
 - Definition von Zwangsbedingungen zur Einschränkung des Wertebereichs von Variablen
 - Überprüfung nur an bestimmten Stellen möglich (keine permanente Überwachung)
 - Überprüfung in Simulink mittels „Model Verification“-Blöcken
- **Vor- und Nachbedingung:**
 - Zwangsbedingungen als Vor- und Nachbedingung von Operationen
 - Aufspalten in Simulink-Subsysteme für Vor-/ Nachbedingung und Operationen
 - Überprüfung in Simulink mittels „Model Verification“-Blöcken





- **Zeitbedingungen:**
 - Definieren von Zeitmarkierungen im Sequenzdiagramm
 - Einschränken der Abhängigkeit zwischen Zeitmarkierungen
 - Überprüfung der Zeitmessung in Simulink mittels „Model Verification“-Blöcken
- **Dateninvarianten:**
 - Definition von Zwangsbedingungen zur Einschränkung des Wertebereichs von Variablen
 - Überprüfung nur an bestimmten Stellen möglich (keine permanente Überwachung)
 - Überprüfung in Simulink mittels „Model Verification“-Blöcken
- **Vor- und Nachbedingung:**
 - Zwangsbedingungen als Vor- und Nachbedingung von Operationen
 - Aufspalten in Simulink-Subsysteme für Vor-/ Nachbedingung und Operationen
 - Überprüfung in Simulink mittels „Model Verification“-Blöcken





- Einführung
- Übersetzen von UML-Komponentendiagrammen nach Simulink
- Übersetzen von UML-Sequenzdiagrammen nach Simulink
- **Zusammenfassung**



- **Übersetzung von Statischen UML-Komponentendiagrammen**
 - Generische UML-Komponentenstruktur
 - Übersetzen von UML-Komponenten nach Simulink
 - Zeitmessung zur Realisierung/Überprüfung von periodischen, aperiodischen und sporadischen Komponenten
- **Übersetzung von Dynamischen UML-Sequenzdiagrammen**
 - In dieser Arbeit immer als Ergänzung zum UML-Komponentendiagramm verwendet
 - Signal und Kontrollfluss
 - Beschränkung von Operationen und Zeitintervallen
 - Dateninvarianten
 - Vor-/Nachbedingungen

