Ostfalia
Hochschule für angewandte Wissenschaften

Wolfenbüttel & Braunschweig

Co-Simulationsbackplane XIL

.Technische Daten und Anwendungsmöglichkeiten

CHANCEN DER CO-SIMULATION

Eine Co-Simulationsbackplane, welche gleichermaßen die Kopplung diverser Modellierungstools, diverser Modellierungssprachen und die Kopplung in Echtzeit darstellt, stellt im gesamten Entwicklungsprozess einen durchgängigen Lösungsansatz zur Verfügung. Mit solch einer Co-Simulationsbackplane kann bereits früh im Entwicklungsstadium simuliert werden, die Modelle passend zum Entwicklungsfortschritt ausgetauscht werden und die dabei entstandenen Modelle und Testfälle beim abschließenden HIL Test wieder verwendet werden.

Erste Modelle werden im Zuge der modellbasierten Entwicklung zunehmend in UML oder ähnlichen Sprachen modelliert. Die Backplane ermöglicht die frühzeitige Absicherung dieser Modelle bspw. gegen Simulink Modelle und andere Modelle aus frühreren Entwicklungsprozessen. Mit Fortschreiten des Entwicklungsprozesses werden entsprechend dem Entwicklungsstand die einzelnen Modelle gegen detailliertere resp. zielorientierte Modelle ausgetauscht. Ein bedarfsgerechter Wechsel der Modellierungssprache ist ganz im Sinne der Co-Simulation. Vorhandene Modelle können weiter genutzt werden. Damit entfällt ein großer Dokumentationsaufwand und die Gefahr der Fehlinterpretation der Dokumentation wird verringert. Die Modellierer können mit ihren gewohnten und auf das Problem optimierten Tools arbeiten. Durch die Echtzeitfähigkeit können bereits auf dem Labortisch erste Muster getestet werden, welche dann später auf einer Prototypenelektronik im realen Umfeld erprobt werden.

Über das XCP-Protokoll kann über eine einheitliche Schnittstelle in alle Modelle gemessen, aber auch kalibriert und stimuliert werden. Wird letzteres Feature genutzt, so besteht neben der vom Modell getrennten Archivierung der Testfälle auch die Möglichkeit, die Testfälle im aufsteigenden Ast des V-Modells weiter verwenden zu können. Ein Neuaufsetzen der Testfälle entfällt.

Eine Co-Simulationsbackplane stellt somit eine Integrierte Entwicklungsumgebung dar, welche den gesamten Entwicklungsprozess vereinfacht, den Dokumentationsaufwand verringert, schnellere Entwicklungszyklen ermöglicht und das Neuaufsetzen von Blöcken vermeidet.

Eigenschaften

- Echtzeitsimulation mit Zyklusraten von 500µs (Jitter: ± 5ns) (Dual Core 2GHz)
- Verteilung der Runables mit unterschiedlichen Zyklusraten auf die Cores der Prozessors
- Kleiner Simulationskernel, dadurch hohe Simulationsperformance
- Lauffähig auf gängiger PC-Plattform (Notebook, Desktop-Systemen, Industrierechnern)
- Lauffähig auf Prototypenelektronik (z.B. Protronic von AFT, Digi-CC-W9P-9215)
- Kostengünstiger Labor-HIL Arbeitsplatz darstellbar
- Bypassing von Funktionen
- PC-Plattformunabhängige Ansteuerung von Analoger-/Digitaler Eingabe/Ausgabe (Unterstützt derzeit ca. 400 verschiedene DAQ Karten von 25 Herstellern)
- Konfiguration über Enterprise Architect (Componenten- / Deployment- Diagramm)
- Code Generierung über Enterprise Architect (Class-Diagramm)
- Visualisierung, Stimulation und Kalibration der Daten über XCP Protokoll

Modellierungssprachen

- Simulink / Stateflow RTW (Echtzeit)
- Simulink / Stateflow
- ASCET-MD
- C-Code
- DAQ Karten (ca. 400 verschiedene Karten von 25 Hertellern)
- Physik-Engine (aus Blender heraus)

.

Ostfalia
Hochschule für angewandte Wissenschaften

Wolfenbüttel & Braunschweig

ARBEITSWEISE

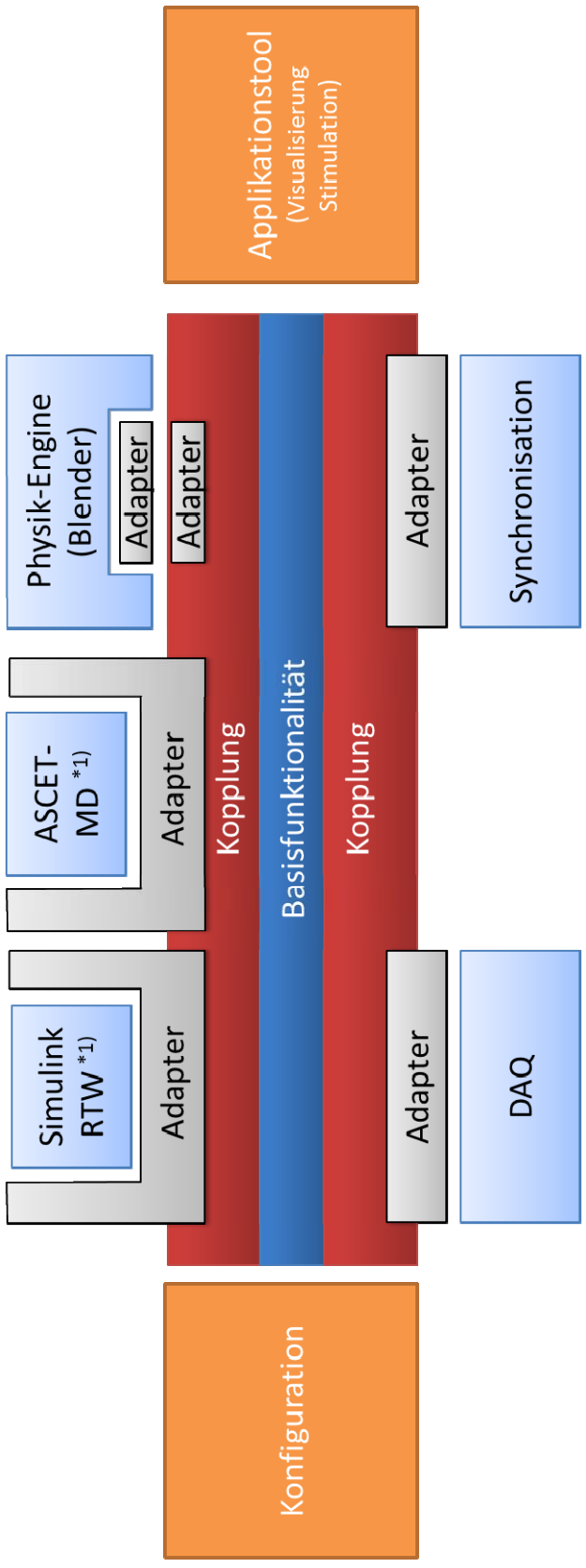
Die Co-Simulationsbackplane ist als ein simulations- und plattformunabhängiger Koordinator ausgelegt. Sie basiert nicht auf der Grundlage einer Simulationsengine und bindet weitere Simulatoren ein bzw. ermöglicht durch Library Funktionen die Erweiterung einer Sprache und weitere Features. Vielmehr wird hier eine tatsächliche Simulatorkopplung durchgeführt, indem die Backplane Ein- und Ausgangssignale der unterschiedlichen Blöcke miteinander koppelt. Die Backplane ist somit ein Bindeglied zwischen den verschiedenen Welten. Sie bildet die unterschiedlichen Schnittstellenfunktionalitäten der einzelnen Simulatoren auf eine Grundmenge ab und verbindet die Simulatoren über diese Basisfunktionalitäten.

Die entwickelte Co-Simulationsplattform besteht aus einer Kernkomponente, welche mit Hilfe von Adaptern erweitert wird. Die Kernkomponente übernimmt alle notwendigen Aufgaben für das Hochfahren, die Steuerung der Simulation, den Datenaustausch, die Messdaten-erfassung, die Fehlererkennung und die Fehlerbehandlung zwischen den Simulationsblöcken.

Die Simulationsblöcke selbst werden mit Adaptern an den Kern angebunden. Die Adapter werden für den jeweiligen Einsatzzweck entsprechend entwickelt und ermöglichen einen Datenaustausch von der Modellierungssprache bzw. des Entwicklungswerkzeuges zum Backplanesystem. Hierbei können zwei Arten von Adaptern unterschieden werden. Zum Einen werden aus C/CPP-Code Laufzeitbibliotheken (.so resp. .dll) erstellt und angebunden (Ladbare Librarys). Zum anderen kann in Simulatoren eine spezielle Library (Integrierbare Library) eingebunden werden, welche für den Datenaustausch mit dem Kern sorgt. Der erstere Adapter findet Verwendung bei codeerzeugenden Modellierungssprachen und Echtzeitanwendungen. Der zweite Adapter ist für Modellierungssysteme gedacht, welche keinen Codenergator (z.B. Matlab oder Blender) besitzen.

Alle Adapter erhalten neben der Kommunikations-schnittstelle auch eine Schnittstelle zur Messdaten-erfassung auf Basis des XCP Protokolls. Diese ermöglicht gleichermaßen das Messen, das Stimulieren und das Anpassen von Verstellparametern.

Die Beschreibung des gesamten Simulationssystems wie z.B. die Simulationsdatenstruktur und die Aufteilung der Komponenten wird in einer globalen XML Konfigurationsdatei beschrieben. Mit Hilfe dieser Datei werden die Adapter dynamisch beim Start der Co-Simulationsbackplane an das Simulationssystem angepasst. Die Datei muss entsprechend vor der Simulation erstellt werden und wird bei einer verteilten Simulation an alle beteiligten Systeme verteilt. Die Erstellung der Konfiguration erfolgt über Enterprise Architect auf Basis von stereotyped UML Elementen.



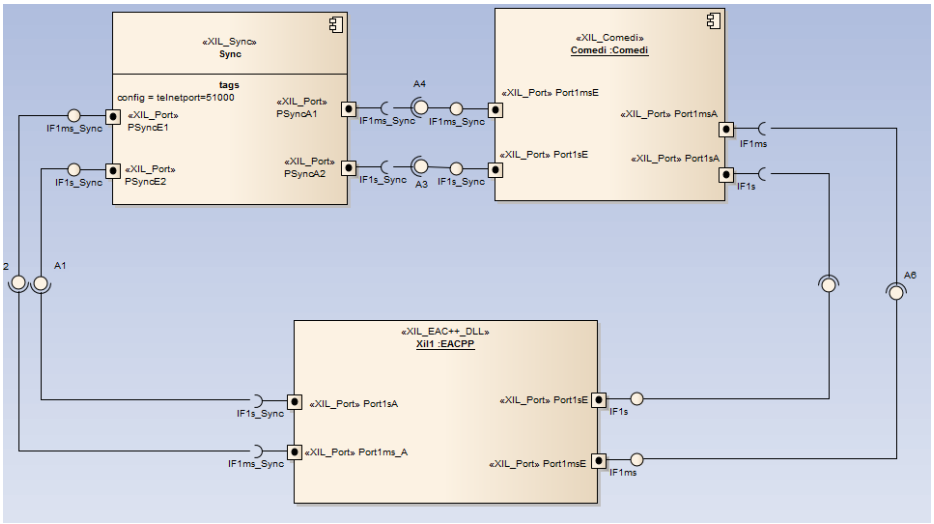
Ostfalia
Hochschule für angewandte Wissenschaften

Wolfenbüttel & Braunschweig

Konfiguration

Die Konfiguration des Systems folgt dem Gedanken des MDA Ansatzes. Das System wird dazu in Enterprise Architect UML konform auf Basis von:

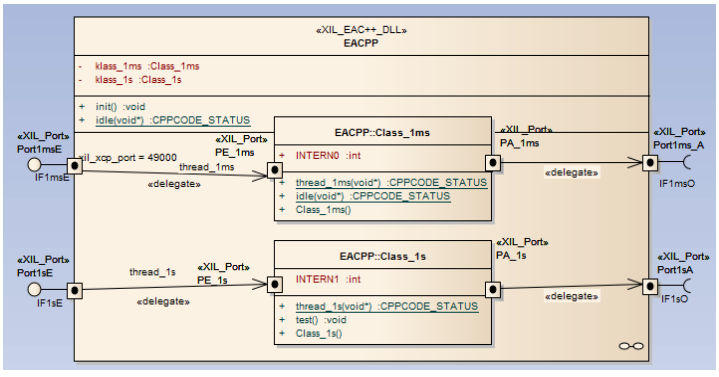
- Component-Diagram für die Zusammenschaltung der Simulationsblöcke,
- Deployment-Diagram für die Verteilung der einzelnen Simulationsblöcke in einem verteilten System resp. auf unterschiedliche Prozesse und
- Class-Diagram für die manuelle Integration von in UML spezifizierten C-/C++Codes und Zustandsautomaten



Mit dem Start der Co-Simulationsbackplane wird auf Basis der generierten Beschreibungsdatei zunächst die Ladbaren Librarys geladen, resp. die Verbindung zu den integrierbaren Librarys hergestellt. Nach anschließender Herstellung der Kommunikationsbeziehungen werden abschließend die notwendigen Threads gestartet. Jeder Event eines Simulationsblockes erhält seinen eigenen Thread. Die Simulation wird wahlweise mit max. Rechengeschwindigkeit, in Echtzeit oder im Single-Step Modus durchgeführt.

Es bleibt dabei den Entwickler überlassen, ob er die Components direkt im Component-Diagram beschreibt, oder diese zunächst in einer Bibliothek erstellt und Instanzen von diesen in das eigentliche Diagramm zieht. Über Import-Funktion können Components auch direkt aus bspw. Simulink Modellen erstellt oder Templates über Export generiert werden. Über diese Methode ist sichergestellt, dass die Konfiguration konsistent zu den externen Simulationsbeschreibungen sind.

Class-Diagrams stellen sicher, das während der CPP-Code Generierung der der notwendige Adapter direkt mit erstellt wird.

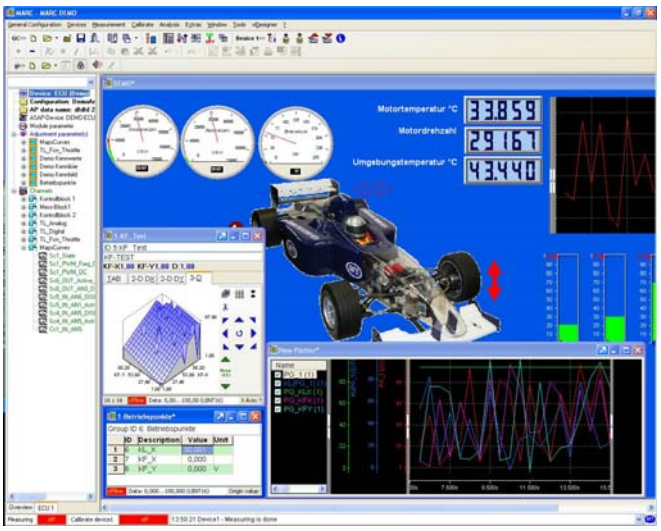


Tool Chain

Zum Starten der Simulation wird zunächst die um die XIL Regeln ergänzte Validierungsfunktion von Enterprise Architect gestartet. Anschließend wird über den XIL CodeGenerator eine Beschreibungsdatei generiert, auf dessen Basis die Co-Simulationsbackplane den Simulationsprozess aufbaut. Sind Class-Diagrams oder andere Components im Simulationsprozess enthalten, die CPP-Code erzeugen resp. Wrapper-Funktionen benötigen, so muss für diese Components ebenfalls der Code-Generator gestartet werden. Über mitgenerierte Makefiles wird für letztere im Anschluss aus dem Source-Code die Ladbaren Librarys erzeugt.

Applikation / Messen

Das Messen von Simulationswerten innerhalb der Simulation und insb. innerhalb der Simulationsblöcke ist einer der wichtigsten Arbeitsschritte während der Simulation. Auf Basis des XCP-Protokolls können beliebige Applikationswerkzeuge (z.B. MARC I) hierfür verwendet werden.

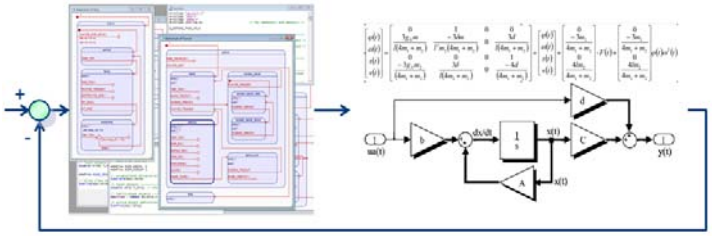


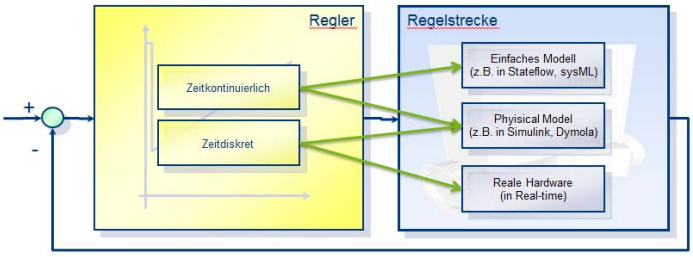
Ostfalia
Hochschule für angewandte Wissenschaften

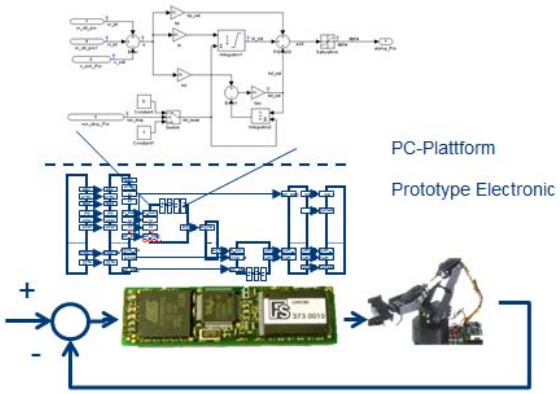
Wolfenbüttel & Braunschweig

EINSATZBEREICHE DER CO-SIMULATION

- Einbindung weiterer (der Aufgabenstellung gerechten) Simulationssprachen und damit Abdeckung eines größeren Einsatzbereiches im Entwicklungsprozess.
- Austausch von einzelnen Simulationsblöcken entsprechend dem Fortschritt des Entwicklungsprozesses gegen detailliertere Blöcke.
- Geringer Dokumentationsaufwand
- Durch die ‚gezwungene' Trennung der Modelle (z.B. in Regler und Regelung) bessere Wiederverwendbarkeit der Blöcke.
- Versionierung der Modelle, der Testfälle und Simulationsantworten.
- Toolunabhängige Einbindung von Hardware (DAQ-Karten) und Entfall der Abhängigkeit an Toolspezifische Hardware
- Integration von Kundenspezifischen Modellen in die eigene Entwurfsmethodik

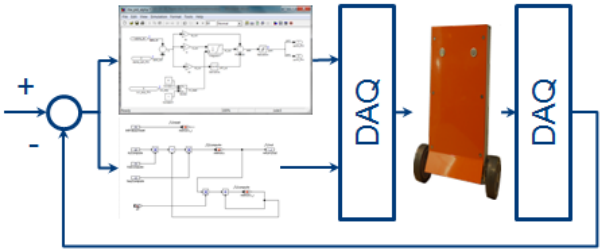


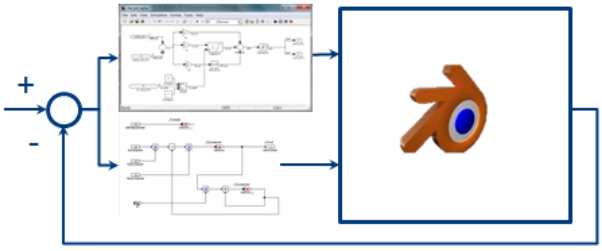




Die Verteilte Co-Simulationsbackplane koppelt unterschiedliche Rechner zu einem großen Rechencluster zusammen. Dadurch können gleichermaßen:

- komplexe Systeme simuliert werden (hohe Simulationsperformance)
- spezifische Systeme angebunden werden (HIL Systeme, spez. Simulationsrechner)
- Prototypenelektroniken mit ausgelagerten Modellierungselementen (Bypassing) gekoppelt werden





Kontakt

Ostfalia Hochschule für angewandte Wissenschaften
Fachhochschule Braunschweig/Wolfenbüttel
Prof. Dr. Detlef Justen
Forschungsvorhaben: X-In-The-Loop*1)
Tel.: 05331 / 939 321 30 Mail: d.justen@ostfalia.de

*1) Gefördert vom BMBF im Rahmen des Programms "Ingenieur Nachwuchs 2008"