DECOMSYS::SIMTOOLS

V Model

- Requirements
- Functional Model
- Application Code
- Validation
- Verification
- Test
V and A Model

- Requirements
- Functional Model
- Application Code
- Test
- Validation
- Verification

A Model

- Architecture-Allocated Functional Model (AAF)
- Functional Model
- Virtual Prototype (VP)
- Middleware Code
- Application Code

DECOMSYS::SIMSYSTEM
DECOMSYS::DESIGNER
DECOMSYS::GENERATOR, DECOMSYS::OILEXPORTER
DECOMSYS::SIMTARGET
DECOMSYS::NODE
AAFM Simulation

Functional Application Task
Simulation in Simulink/Stateflow

Time-Triggered Operating System
Simulation

Simple FT-Com Simulation

Generic Communication without Time Delays

AM

Configuration Simulation Code Generation

FlexRay

Pedal ECU

Control ECU

Wheel ECU

0 100 200 300 400 500 600 700 800 900 1000us
Virtual Prototype

- Simulation
- Code Generation

FlexRay

Wheel ECU

Pedal ECU

Control ECU

0       100     200     300     400     500      600     700    800     900    1000us

FlexRay

Generic Communication without Time Delays

Simulink/Stateflow

Time-Triggered Operating System Simulation

FT-Com Simulation

FTCOM Configuration

DECOMSYS
**SIMSYSTEM Building Blocks**

- **Cluster**: connection to XCDEF file
- **Signal Connectors**: signal transmission
- **Tasks**: time-triggered functional model
- **Hosts**: hosting Microprocessors

**AAFM Example**
Communication Fault Injection

Functional Application Task Simulation in Simulink/Stateflow

Time-Triggered Operating System Simulation

FT-Com Simulation

Simulation-Based Communication Fault Injection

How the application reacts on communication faults?

- Simulation based fault injection
  - Spontaneous injection of communication faults during simulation
    - Uses Graphical User Interfaces
  - Fault scenario modelling in Matlab/Simulink
    - Modelling interfaces to FlexRay simulation core
Spontaneous Comm. Fault Injection

Comm. Fault Scenario Modeling
Source Code Generation

Rational
- Application code generation for each Host
- Task bodies generated from SIMSYSTEM Task blocks
- Interfacing with FTCom source code and OIL task calls
- Supporting Rapid Prototyping and HIL
- Supporting prediction code generation
- Upload of binary to target hardware
- Start and stop the application on target hardware

Source Code Generation

Supported code generators
- Real-Time Workshop
  - SIMTARGET<...,Linux>
  - DECOMSYS Prototyping Platform with Linux based OS
  - Customer specific platforms
  - dSPACE RTI FlexRay Blockset Integration

- Real-Time Workshop Embedded Coder
  - SIMTARGET<...,TimeCore>
  - Compatible with TimeCore production software

- Target Link
  - SIMTARGET planned on customer request
Source Code Generation Example

Virtual Prototype (VP)

SimTARGET

TimeCore

Architectural Model (AM)

DESIGNER GENERATOR OILEXPORTER

Customer ECU Hardware

Source Code Generation Example

FlexRay Restbus-Simulation

Physical Process

ECU

I/O CAN I/O CAN I/O CAN I/O CAN

Task1 Task2

FTCom FlexRay

Restbus-Simulator

Task3 Task4 Task5 Task6 Task7 Task8

FTCom FlexRay FTCom FlexRay FTCom FlexRay

Supplier 1 Supplier 2 Supplier 3 Supplier 4
MB FlexRay Development

Summary

- Designing multi ECU models
- Critical role of communication subsystem
- Virtual prototyping including communication
- Model based fault injection and simulation
- Cost efficient prototypes
- Early identification of erroneous distributed system behaviour
- Communication system evaluation and training
- Code generation interfaces
- Easy configuration of rest system simulation

Thank you for your attention!

© DECOMSYS 2004