



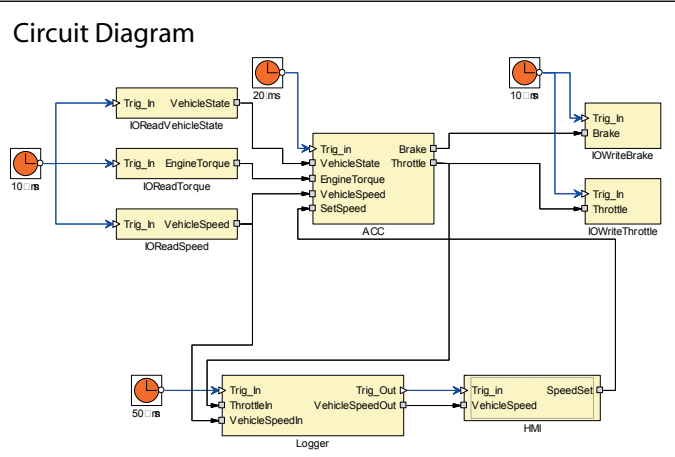
The Rubus[®] Component Model for Resource Constrained Real-Time Systems

Arcticus Systems

Rubus methods and tools supporting Model Based Development (MBD) focus on development of dependable embedded real-time systems that put requirements on analysis and synthesis as well as on design.

The Rubus Component Model (*Rubus CM*) is designed for the development of resource constrained real-time systems. The model aims at supporting three important activities:

- Design
- Synthesis
- Analysis

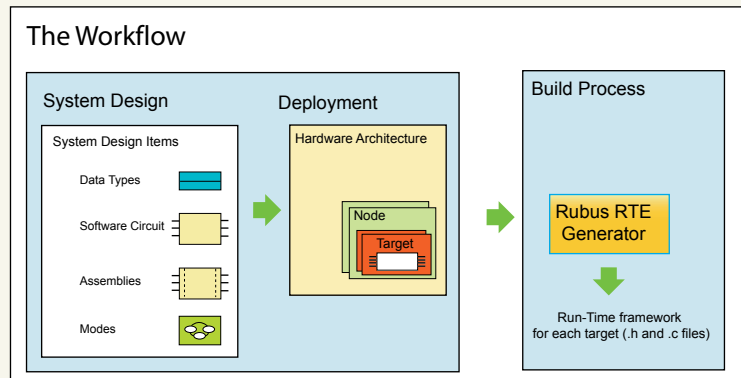


Design

The composition of system consists of application and basic software functionality implemented as SW circuits communicating via ports organised in Circuit Diagrams in analogy with electronic schematics.

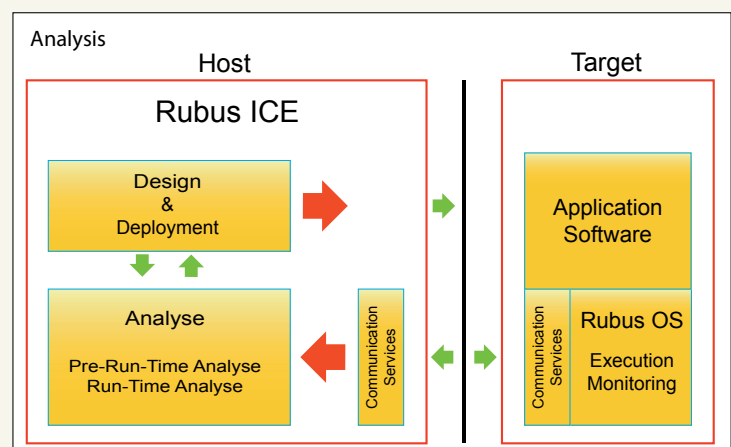
Synthesis

The Deployment process distributes the SW items on a selected HW architecture consisting of the nodes and targets. The Build process generates the run-time framework of each target.



Analysis

To formally verify the model, response time analysis is performed based on the given real-time properties of the SW items. For Timed Triggered items a schedule is generated for each mode. The Run-Time properties of the SW items are monitored and fed back to the model.



Arcticus Systems AB

Box 530, Datavagen 9A
S-17526 JARFALLA, Sweden

www.arcticus-systems.com
info@arcticus-systems.com

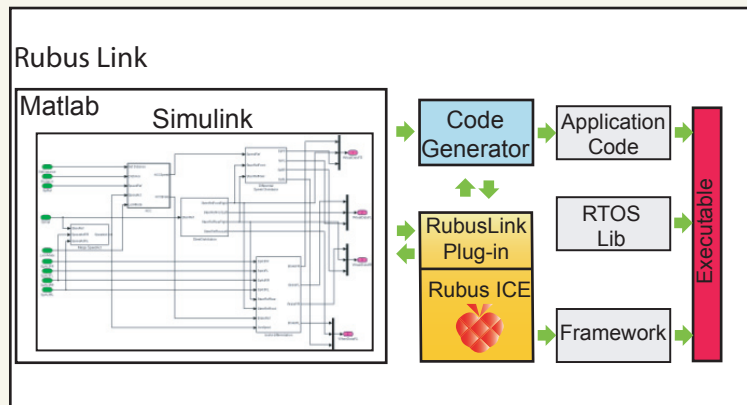
© 2008, Arcticus Systems AB, All rights reserved

Rubus[®] is a registered trademark of Arcticus Systems AB

Integration Matlab/Simulink

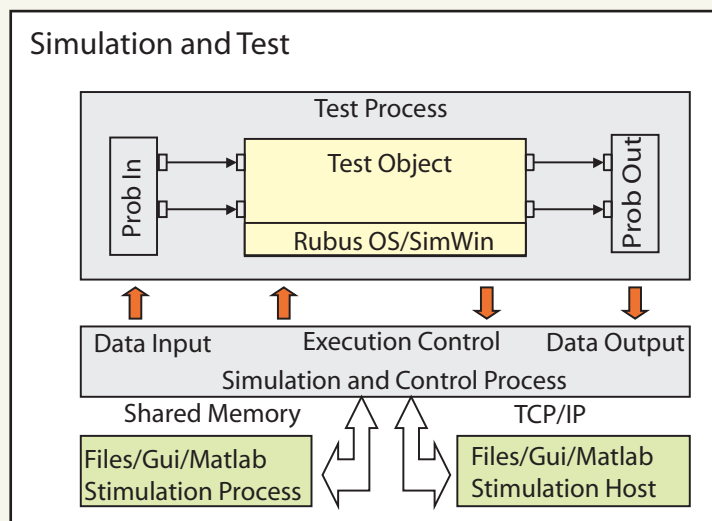
A SW item can be implemented using Simulink. By adapting the Simulink code generator to match structures required by the Rubus ICE, no wrapper functions are needed.

The integration of Simulink models in the Rubus CM facilitates the real-time analysis of the entire design.



Simulation and Testing

To verify the logical functionality of SW items a simulation framework is supported. The framework reads data from external tools or files as Matlab feeds the simulation process stimulating the input ports and state variables. The output from the simulation process is fed back to the external tools. This gives developers possibility to test the logical functionality.



Autosar

"Autosar has focused on location independence, interface standardization, and code portability, but not on software quality (timings and dependability)." IEEE Computer, October 2007.

The design of Rubus CM has been influenced by Autosar and can be seen as a subset of the Autosar Component Model, with support on software quality.

In the Rubus CM the design of the entire software of a target is supported including Basic Software modules. This objective is a prerequisite of analysability.