Scilab/Scicos Code Generator for Flex

Mauro Marinoni [nino@evidence.eu.com]
Overview

- Scilab/Scicos overview
- Code Generator
  - Flex Version
- Demos
What is Scicos?

is a scientific software package for numerical computations providing a powerful open computing environment for engineering and scientific applications.

is OS independent:
* Windows *(including Vista)*
* Unix, Linux,
* Mac OSX
* ... your custom O.S. ☺

A number of toolboxes are available with the system:

* 2-D and 3-D graphics
* Linear algebra
* sparse matrices
* Polynomials
* Rational functions
* Interpolation
* approximation
* Simulation: ODE solver and DAE solver
* Scicos: a hybrid dynamic systems modeller and simulator
* Classic and robust control, LMI opt.
* Signal processing
**Scicos** is a graphical dynamical system simulator toolbox built in [Scilab](https://www.scilab.org). With **Scicos** you can create block diagrams to model and simulate the dynamics of hybrid systems, control real system in real time with Scicos Hardware In the Loop (Scicos-HIL) and compile your models into executable code for faster simulation and *stand alone embedded applications*.

With **Scicos** you can:

- Graphically model, compile, and simulate dynamical systems
- Combine continuous and discrete-time systems
- Simulate digital communications systems with Scicos-ModNum
- Use implicit blocks developed in the Modelica language

**Scicos** is used for signal processing, control systems, queuing systems, and to study physical and biological systems.

New extensions allow generation of component based physical modelling of electrical and hydraulic circuits using the Modelica language.

Planned major technical developments:

- Graphics, Scilab GUI and GUI builder
- Scicos industrialization (GUI, quality,...)
- Documentation
- New kernel, 64 bits and 128 bits technology
- Improvement and updating of algorithms: control, signal processing, identification,...

Excellence domains:

- Interoperability (with standard scientific software) and services architecture
- HPC (High Performance Computing), Grid Computing, parallel computing, multi-core
- C code generation, embedded systems
- R & D: developments in collaboration with research

Scilab 5.0 (2008):

- New license: CeCILL (GPL2 compatible) and GPL2.
- Modularization
- New and graphics rendering GUI
What do you need for your embedded RT applications?

- **Faster**
  - Reduced development time = Minimum Time To Market
  - Better tools
  - Access to source code and development chain

- **Better**
  - High quality, flexibility, market superiority
  - Knowledge
  - Collaboration
  - Independence

- **Cheaper**
  - Competitive
  - No patent
  - No royalties
  - No hidden costs
  - Protected by OS licenses
For modeling and Simulation

- Scilab: Scilab language (script)
- Scilab: integration with other programming languages (C/C++, Java, FORTRAN, etc.)
- Scicos: Scicos diagram (visual programming)
- Scicos: integration with other simulation platform (Modelica, GHDL, etc)

But also for embedded applications

- **Real Time simulation with Real plants**
  - Scicos Hardware In the Loop
    - Scicos-HIL: the Scicos simulator executes the control section in real time and uses data acquisition cards for the connection with the real plant

- **Code Generation from Scicos diagram**
  - Scicos internal GP code generator
  - Scicos-RTAI: for Linux RTAI systems
  - Scicos-FLEX: for micro controllers and DSPs
**Code Generation with Scilab/Scicos**

Scicos functional modeling

INRIA/SUPSI Code Generator

Simulation

Same Behavior!

HW + Erika Enterprise
- It converts a **Scicos** superblock into a **dsPIC** application ready to be executed by the MCU.
- It supports single-rate single-task controller.
- *One and only one* source of time is required.
A set of palette for the specific Hardware has been produced.
The **Scicos superblock** is mapped to an **Erika periodic** task.

The task is executed with a **period** which is equal to the **Scicos Timesource period**.

Some tricks are needed in order to improve performances.
To each block is connected a function

- **Init**, **InOut** or **Close** depending on the system status

The application is executed block by block (function by function) following the **“data path”**
Some parameters are required by the generation engine.
Mauro Marinoni

[nino@evidence.eu.com]

- **Retis:**
  - [http://retis.sssup.it](http://retis.sssup.it)

- **Evidence:**
  - [http://www.evidence.eu.com](http://www.evidence.eu.com)

- **Scilab:**
  - [http://www.scilab.org](http://www.scilab.org)

**RETIS Lab**
Real-Time Systems Laboratory