Context

Goal: export a structural[1] view of a SR[2] functional model to Eclipse, as EMF model

[1] Subsystems, topology of communications, rate constraints, partial order of execution contraints, . . .

[2] Synchronous-Reactive MoC as in Simulink/Scicos/XCos

Context

Motivations

- Generation of behavioral code (Evidence's E4Coder for Scicos/XCos, avail. at `http://www.e4coder.com/`)
- System level design with separation of the functional and platform models
- Generation of flow-preserving implementations for multi-core (distributed) platforms
- Generation of models that include computation and communication delays (control, scheduling, network multi-sim.)
- Synthesis and optimization of SW configuration (tasks & messages)

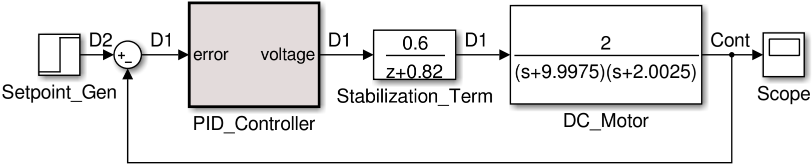
⇒ EMF-based meta-model for Simulink (Scicos/XCos)

Context

Related Work

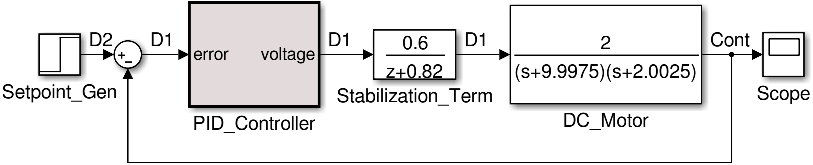
- Gene-Auto (ITEA project 2005-08)
- Project P (FUI project 2011-14)
- QGen (instantiation of the technology developed within Project P in a commercial product — a qualifiable ISO 26262, DO-178B/C code-generator and model verifier, released in Q4 2014 by AdaCore)
- Massif (open-source, released in Q4 2014)

Simulink Concepts



Which are the primary entities (and their attributes/relationships) representing a Simulink model?

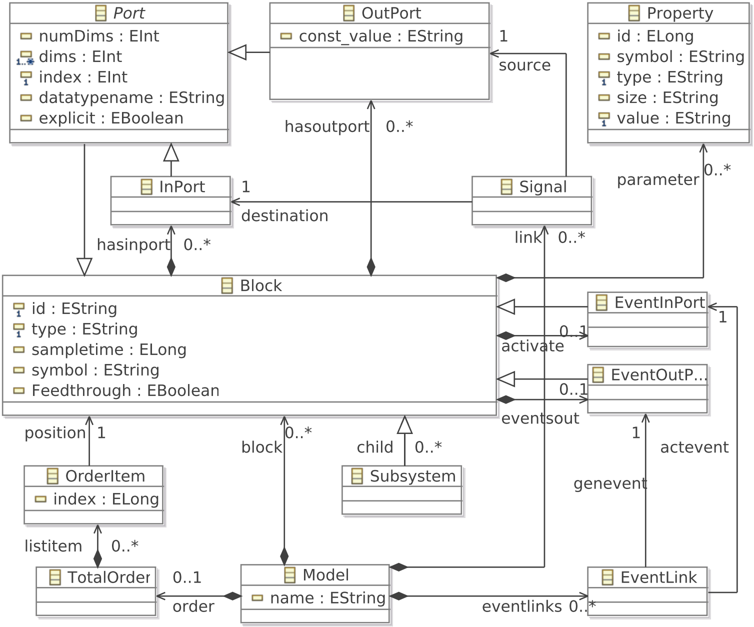
Simulink Concepts



Which are the primary entities (and their attributes/relationships) representing a Simulink model?

- A model is a network of connected blocks (*blocks, ports, links*)
- Some blocks add hierarchy to organize a diagram (*subsystems*)
- Blocks are configured through *parameters*
- Blocks may be executed at specific rates (*sample time*)
- Blocks output may depend on inputs (*direct feedthrough*)
- Subsystems may react to triggers (*event ports & links*)
- Blocks execute according to a *total order of execution*

An EMF-based meta-model for Simulink (Scicos/XCos)



The slx2emof Model Exporter

Features

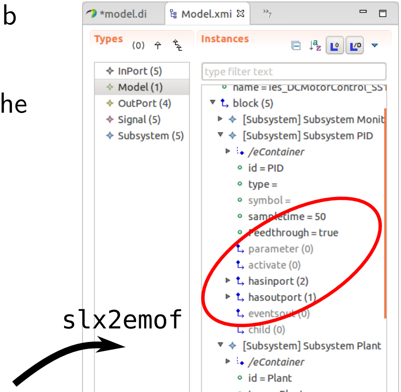
- Simple implementation in Matlab

Usage restrictions

- Subsystems at the top level of the model hierarchy

Limitations (WiP)

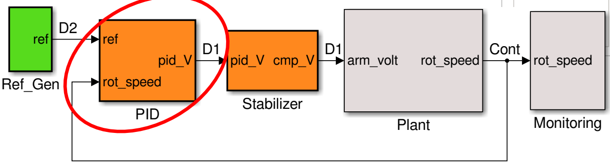
- Doesn't cover all the concepts in the meta-model
- Not all the models work, even for the covered concepts



- InPort (5)
- Model (1)
- OutPort (4)
- Signal (5)
- Subsystem (5)

```
name = Tes_DCMotorControl_SS1
▼ block (5)
  ▶ [Subsystem] Subsystem Monit
  ▼ [Subsystem] Subsystem PID
    ▶ /eContainer
      id = PID
      type =
      symbol =
      sampletime = 50
      feedthrough = true
      parameter (0)
      activate (0)
    ▶ hasinport (2)
    ▶ hasoutport (1)
      eventsout (0)
      child (0)
  ▼ [Subsystem] Subsystem Plant
    ▶ /eContainer
      id = Plant
      type = Plant
      symbol =
      sampletime = 0
```

slx2emof



The slx2emof Model Exporter

The Matlab code is based on the *Matlab-XML export API* and the Matlab's model construction commands

```
% Create the document node and root element
doc_node = com.mathworks.xml.XMLUtils.createDocument('
    ↪ com.eu.evidence.functional:Model');

% Identify the root element, set the version attribute
ee_node = doc_node.getDocumentElement;
ee_node.setAttribute('xmi:version','2.0');

% Set other attributes ('xmlns:xmi','xmlns:xsi','xmlns:
    ↪ com.eu.evidence.functional')

% Set the name of the model (gcs)
ee_node.setAttribute('name', gcs);

% ... POPULATE THE MODEL HERE ...

% Export the DOM node to Model.xmi
xmlwrite('Model.xmi',doc_node);
```

The `slx2emof` Model Exporter

The Matlab code is based on the *Matlab-XML export API* and the Matlab's model construction commands

Produces

```
<?xml version="1.0" encoding="utf-8"?>
<com.eu.evidence.functional:Model
  xmlns:com.eu.evidence.functional= "http://www.evidence
      ↪ .eu.com/functional"
  xmlns:xmi="http://www.omg.org/XMI"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  name="ies_DCMotorControl_SS1" xmi:version="2.0">

...

</com.eu.evidence.functional:Model>
```

The slx2emof Model Exporter

The Matlab code is based on the *Matlab-XML export API* and the Matlab's model construction commands

```
% For each subsystem
for i = 1:length(ssHdls),
 % Add the block element node
 ss_node = doc_node.createElement('block');

 % Assign the right xsi:type attribute
 ss_node.setAttribute('xsi:type','com.eu.evidence.
     ↪ functional:Subsystem');

 % Set the sampletime attribute
 sampletInfo = get_param(ssHdls(i),'CompiledSampleTime');
 ss_node.setAttribute('sampletime', num2str(1000*
     ↪ sampletInfo(1)));

 % Set all the other attributes, ports,
 % the port-connectivity descriptions, etc.
end
```

The `slx2emof` Model Exporter

The Matlab code is based on the *Matlab-XML export API* and the Matlab's model construction commands

Populates the model with, e.g., this code

```
<block Feedthrough="true" id="PID" sampletime="50" type=
    ↪ "" xsi:type="com.eu.evidence.functional:Subsystem
    ↪ ">
  <hasinport datatypename="double" id="PID_ref" index="0
      ↪ " numDims="1">
    <dims>1</dims>
  </hasinport>
  <hasinport datatypename="double" id="PID_rot_speed"
      ↪ index="1" numDims="1">
    <dims>1</dims>
  </hasinport>
  <hasoutport datatypename="double" id="PID_pid_V" index
      ↪ ="0" numDims="1">
    <dims>1</dims>
  </hasoutport>
</block>
```

The slx2emof Model Exporter

The Matlab code is based on the Matlab-XML export API and the *Matlab's model construction commands*

```
% Find systems, blocks, lines, ports, ...
ssHdls = find_system(mdlHdl,'SearchDepth',1,'BlockType',
    ↪ 'SubSystem');
inPortBlkH = find_system(ssHdls(i),'SearchDepth',1,'
    ↪ Blocktype','Inport');
outPortBlkH = find_system(ssHdls(i),'SearchDepth',1,'
    ↪ Blocktype','Outport');

% Get name/value of the specified parameter for a block
get_param(inPtHdls(p),'PortNumber'),
ssPrtCnInfo = get_param(ssHdls(i), 'PortConnectivity');

% Execute particular phase of simulation of model
eval([mdlStr,'([], [], [], ''compile'');']);
sampletInfo = get_param(ssHdls(i),'CompiledSampleTime');
cpDms = get_param(inPtHdls(p),'CompiledPortDimensions');
```