Workshop 1:
System Design in Avionics & Space

Specification for SystemC-AADL interoperability

Eugenio Villar
Eduardo de las Heras
Microelectronic Engineering Group
University of Cantabria
Outline

• Motivations
• General Concepts
  – AADL
  – SystemC
  – PERFidiX and SCope
• AADL-SystemC Design Flow
• Mapping AADL to SystemC
• Example
Motivations

• System design issues:
  – Incomplete capture of specification
  – Need for design refinement and validation
  – Impact of functional and non-functional properties
    • Timing properties
    • Platform architecture

• Software/Hardware co-design
AADL Concepts

PIM

processor binding

thread 1

connection binding

memory binding

thread 2

data

processor binding

processor1

bus

memory

device

processor2
SystemC Concepts

- **SystemC features**
  - Standard platform for system design (IEEE 1666) developed by the OSCI
  - C++ extension
  - Strict-time, event driven simulator
  - Concurrent Execution Kernel
SystemC Concepts

• SystemC Basic Elements
SCoPE Concepts

• System Co-simulation and Performance Estimation in SystemC
  – Multi-processor SW source-code simulation
    – OS Modelling
      – POSIX
    – Timed SW simulation
    – Performance estimation of SW code
      – Time & Power

www.teisa.unican.es/scope
SCoPE Concepts

PIM

Application Code
Task 1  ...  Task n

POSIX API
Packages
PERFidiX
Drivers

Memory
Proc. 1  Proc. 2  Proc. n

Bus 1

Devices

PSM

Memory

Bus n

Devices

Devices
AADL-SystemC Design Flow

- AADL Application Model
- AADL Execution Platform Model

AADL To SystemC translation

Platform independent SystemC description

AADL To Scope parameters translation

SystemC Platform model

Binding

SystemC executable model

Performance analysis

Simulation

Configuration parameters

SCope
AADL to SystemC Framework

ECLIPSE

OSATE

Graphical editor

Textual editor

AADL Model

Refinement

SC HW description

C/C++ Code

XMI file

SystemC integrator and generator

SCope

Configuration parameters

Simulation

Performance analysis

HetSC

SCV

SC HW description

C/C++ Code

XMI file
AADL Semantics in SystemC

AADL

Thread: Schedulable unit of sequential source code.
- Properties
  - Dispatch protocol
  - Period
  - Deadline

SystemC

SC_THREAD: Is called once when simulation Start.
- Properties
  - Specific SC_THREAD implementation
  - SC_TIME, wait (SC_TIME)
  - Assertions SCV
AADL Semantics in SystemC

**AADL**

**Process**: space partitioning where protection is provided

**Subprogram**: sequentially executable source text

---

**SystemC**

**SC_MODULE**: principle structural building blocks of SystemC

**C++ function**: called from the SC_THREAD
**AADL Semantics in SystemC**

**Data**: Enable manipulate data in concurrently in non-deterministic order.
- Properties
  - Concurrency_Control_Protocol

**Channel**: Enable communication between modules
- Properties
  - Semaphores, mutex, custom channels.
Processor: Abstraction of hardware and software responsible for scheduling and executing threads.

- Properties
  - Process_Swap_Execution_time
  - Thread_Swap_Execution_time
  - Scheduling_Protocols

High level, POSIX simulation library and performance Analysis

- SCope configuration parameters
- POSIX scheduling_protocols
AADL Semantics in SystemC

**AADL**

- **Memory**: platform component that stores binary images.

**SystemC**

- **Bus**: platform component that can exchange control and data between modules.
  - Properties
    - Transmission time, propagation delay

**System Co-simulation and Performance Estimation in SystemC**

**SCope configuration parameters**
AADL Semantics in SystemC

**AADL**

- **Devices:** Execution platform component that interface with the exterior
  - Event data port
  - Event port
  - Data port

**Ports and Connections:** Logical Connections to exchange control and data between threads.

**SystemC**

SystemC description at various levels:
- TLM
- RTL
- Synthesis

<table>
<thead>
<tr>
<th>Signal channel, ports, interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIFO channel ports, interface</td>
</tr>
<tr>
<td>Custom Channels, ports, interface</td>
</tr>
</tbody>
</table>
Example

```c
SC_MODULE(wises_example_system_example_impl_Instance) {
    process_producer_0 *process_producer;
    process_consumer_0 *process_consumer;

    sc_fifos::connection_1;
    sc_fifos::connection_2;

    SCCTOR(wises_example_system_example_impl_Instance)
    {
        process_producer = new process_producer_0("process_producer");
        process_consumer = new process_consumer_0("process_consumer");

        process_producer->connection_1[connection_1];
        wises_example_system_example_impl_Instance->connection_1[connection_1];
        process_producer->connection_2[connection_2];
        wises_example_system_example_impl_Instance->connection_2[connection_2];
    }
};
```
Example

```
SC_MODULE(process_producer)
{
    void thread_producer();
    sc_fifo_out<int> port_1;
    sc_fifo_out<int> port_2;

    SC_CTOR(process_producer)
    {
        SC_THREAD(thread_producer);
    }
}
```
Example

Refinement

```c
void process_producer::thread_producer()
{
    while(true)
    {
        //C or C++ code implementing the
        //software functionality
        sc_time period_time(100,SC_MS);
        wait(period_time);
    }
}
```
Conclusions

• SystemC allows modeling AADL
  – Different abstraction levels
  – Refinement
  – Validation

• Specification for model transformation from AADL to SystemC

• Tool proposal for embedded system design
THANK YOU FOR YOUR ATTENTION

QUESTIONS ?