Specification for SystemC-AADL interoperability

Eduardo de las Heras Palmero
Microelectronic Engineering Group
University of Cantabria
• Motivations
• General Concepts
  – AADL
  – SystemC
  – PERFidiX and SScope
• 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 until the system integration
- Timing properties
- Software/Hardware co-design
AADL Concepts

Architecture Analysis & Design Language

- Standard by the SEI, November 2004
- Graphical and textual Language
- Architecture and model based design
- Precise syntax and semantics
- Specification of Tasks and communications
- Enable analysis and validation of constraints
- Large-scale architectures in a single model
- Incrementally refined
- Analyze the system structure and runtime behavior
AADL Concepts
SystemC Concepts

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

• SystemC Basic Elements
SCope Concepts

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

www.teisa.unican.es/scope
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

Simulation

Performance analysis

Configuration parameters

SCope
AADL to SystemC Framework

ECLIPSE

OSATE

Graphical editor

AADL Model

Textual editor

XCMI file

SC Hw description

C/C++ Code

XCMI file

Refinement

SystemC integrator and generator

XC Hw description

C/C++ Code

XC Hw description

HorSC

SCV

SC Hw description

SystemC integrator and generator

SC V

SC Hw description

Configuration parameters

Simulation

Performance analysis

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

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

**AADL**

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

**SystemC**

→ **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.

**Bus**: platform component that can exchange control and data between modules.

- Properties
  - Transmission time, propagation delay

**SystemC**

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

Signal channel, ports, interface

FIFO channel ports, interface

Custom Channels, ports, interface
AADL Semantics in SystemC

AADL

- Flow source
- Flow path
- Flow sink

Corresponding access to subcomponents involve in the flow implementation

SystemC

Flows: support for various forms of flow analysis

- Random generation of tokens using SCV
- Implementation of Write and read access method
- Data recording for posterior analysis using SCV
Example

system_example

```
SC_MODULE(wises_example_system_example_impl_Instance)
{
  process_producer : *process_producer;
  process_consumer : *process_consumer;

  sc_if(is_connection_1)
  {  
    connection_1;
  }
  sc_if(is_connection_2)
  {  
    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->connexion_1|connection_1;
  wises_example_system_example_impl_Instance->connexion_1|connection_1;
  process_producer->connexion_2|connection_2;
  wises_example_system_example_impl_Instance->connexion_2|connection_2;
}
```
Example

process_producer

100ms

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

thread_producer

Refinement

```cpp
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?