

UML and SystemC

a language comparison
for automatic code generation

Per Andersson

Lund Institute of Technology
Sweden

2006-10-17



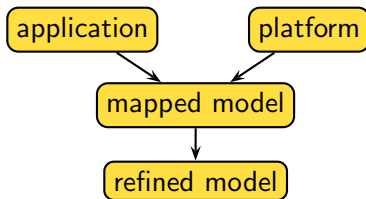
Combine

- model driven architecture
- SystemC
- Y-design flow



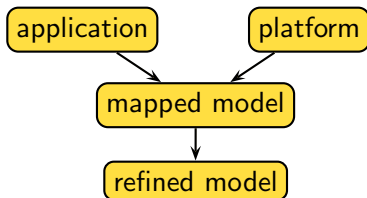
Combine

- model driven architecture
- SystemC
- Y-design flow



Combine

- model driven architecture
- SystemC
- Y-design flow



Lund Institute of Technology

- UML to SystemC translation
- level of abstraction: programmers view (PV)
- case study



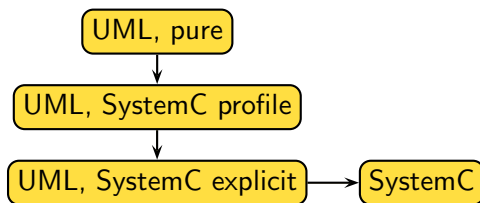
SystemC is C++

extensions:

- discrete event simulator
- macros
- classes
 - module
 - port
 - channel



Transformation in Three Steps



Similarities

- package/name space
- class/module
- port
- interface



UML

- required interface
- realized interface
- interface list
- connector
- implicit message queue



UML

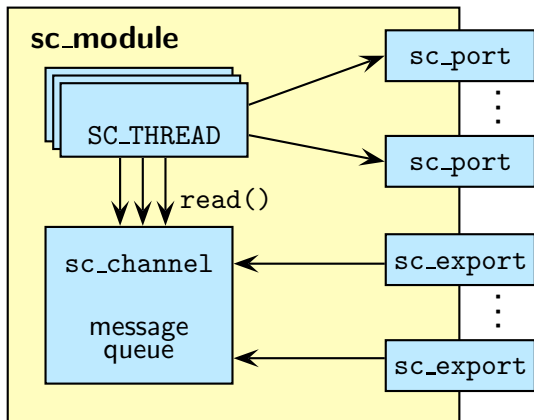
- required interface
- realized interface
- interface list
- connector
- implicit message queue

SystemC

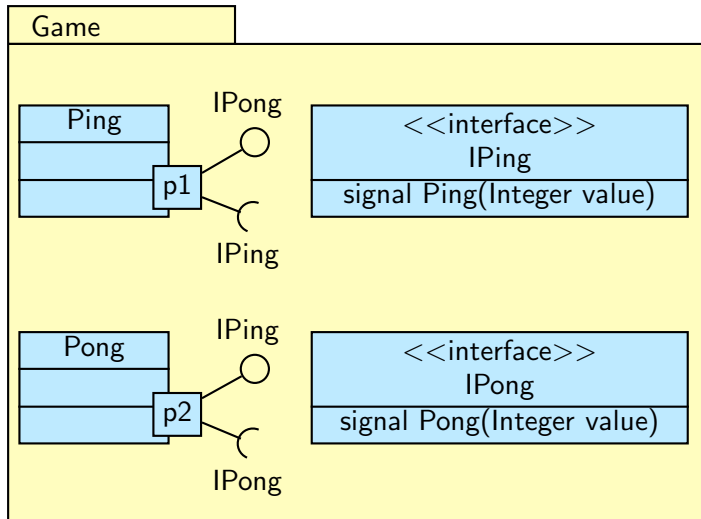
- port
- export
- *one* interface per port/export
- channels implement interfaces
- message queue commonly in the channel



Generated Structure



Example

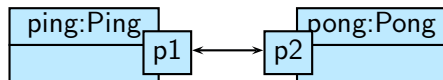


Example

```
class IPing: public sc_interface {
public:
    virtual void Ping(int value) = 0;
    class Ping_signal: public UML_signal {
        public:
            int value;
            inline Ping_signal(int value) :
                value(value) {}
    };
};
```



Example



```
Ping ping("ping");  
Pong pong("pong");  
ping.p1_port(pong.p2_export);  
pong.p2_port(ping.p1_export);
```



Example

```
SC_MODULE( Ping ){
    public :
    sc_export<IPing> p1_export;
    sc_port<IPong> p1_port;
    class Channel_class : public sc_channel ,
                          public IPing{
        private :
        std::deque<UML_signal *> queue;
        sc_event e;
        public :
        Channel_class(sc_module_name name);
        UML_signal *read();
        void Ping(int value);
    };
    Channel_class Ping_channel;
    void Ping_thread();
};
```



Example

```
void Ping::Channel_class::Ping(int value){  
    queue.push_back(new Ping_signal(value));  
    e.notify();  
}
```

```
Ping(sc_module_name name) : sc_module(name),  
                           Ping_channel("Ping_channel"){  
    p1_export(Ping_channel);  
    SC_HAS_PROCESS(Ping);  
    SC_THREAD(Ping_thread);  
}
```



- communication modelling differs
- UML → SystemC mapping is a many to many mapping
- predefined structure for communication
- the SystemC model is *constructed* from the UML model



Thank You

Questions?

