Universal Scripting Interface for SystemC

Rolf Meyer, E.I.S., TU Braunschweig
SoCRocket TLM Models

All models developed with RTL equivalents as blueprint

- Models available in loosely timed (LT), and approximately timed (AT) flavor of TLM2.0.
- ESA Reference TLM Platform
SoCRocket - The building blocks

SoCRocket is more than a model library

 models

foundation

base

blocks

roof

top

Virtual Platforms

Deps Mgr.

Waf Build System

Repo Mgr.

Model Libraries
(Core)

(Media)

(Terma)

IEEE SystemC

IEEE TLM

Cadence scireg

GreenSoCs

GreenReg

ESA TrapGen

GreenSoCs

GreenSockets

Carbon AMBAKit

GreenSoCs GreenControl

c3e SignalKit

© Accellera Systems Initiative
Desired Features

• Automating procedures
• Configuring simulation parameters
• Access runtime simulation information
• Easy testing integration
• Interactive introspection
Desired Features

• Automating procedures
• Configuring simulation parameters
• Access runtime simulation information
• Easy testing integration
• Interactive introspection

→ Perfect tasks for scripting languages
Scripting Language

- Defining variables/option values
- Recording/executing command sequences
- Capturing output results
- Branching and looping
- Importing and exporting options and
- Bridging the gaps to other abstraction levels
Available simulators

- Accellera SystemC
- ALDEC Riviera-PRO → TCL
- Cadence ncsim → TCL
- Mentor Graphics QuestaSim → TCL
- Synopsys Platform Architect → TCL
- ...

- The choice depends on the problem/programmer
Scripting in EDA/SystemC

Most common: TCL
• Command oriented (Shell like)
• No native OOP
• Therefore not the best user experience

Scripting support in Accellera SystemC
• No integrated language
• Different approaches available
Scripting languages for SystemC

Existing implementations

• TCL
• SystemLua → focused on configuration
• GreenScript → focused on abstract modeling

Desired implementation

• depends on the problem/programmer
• we propose Python for its usability
Common SystemC APIs

For example:

• IEEE 1666-2011
• GreenSoCs
• Cadence
• Carbon
• Own
• …
Additional requirements

• Available in multiple Simulators
• Language independent
• Same/similar APIs as in SystemC/C++

• Addressable via hierarchical module name
  
  top.obj.reg
Environment

Scripting Interface

Interface Delegation

Hardware Platform (SystemC/C++)
Environment

Scripting Interface

Interface Delegation

Delegation Kernel

sc_object

SystemC hierarchy

Plug-in (Util) API C++

Hardware Platform (SystemC/C++)
Scripting environment

SystemC

sc_register

constructed
obj.reg.ctrl
Initialization

- Python
- sc_register
- constructed obj.reg.ctrl

SystemC
Python

```
ctrl = usi.find('obj.*.ctrl')
```

**Initialization**

**SystemC**

- `sc_register`
- `constructed obj.reg.ctrl`
Initialization

Elaboration

SystemC

Python

USIDelegate

sc_register

constructed obj.reg.ctrl

ctrl = usi.find ('obj.*.ctrl')

USIDelegate ('obj.reg.ctrl')
Initialization

Elaboration

Scripting environment

Python

USIDelegate

SystemC

sc_register

constructed obj.reg.ctrl

ctrl = usi.find ('obj.*.ctrl')

USIDelegate ('obj.reg.ctrl')

sc_simulation_context

© Accellera Systems Initiative
Initialization

Elaboration

Python

USIDelegate

ctrl = usi.find('obj.*.ctrl')

USIDelegate('obj.reg.ctrl')

Query sc.sim_context for obj.reg.ctrl

sc_simulation_context

constructed

obj.reg.ctrl

ctrl = usi.find('obj.*.ctrl')
Initialization

Elaboration

Python

USIDelegate

sc_register

constructed

obj.reg.ctrl

ctrl = usi.find

('obj.*.ctrl')

sc_simulation_context

USIDelegate

('obj.reg.ctrl')

Query sc_sim_context

for 'obj.reg.ctrl'

Try to convert to

registered interfaces

© Accellera Systems Initiative
Initialization

Elaboration

Python

USIDelegate

sc_register

constructed
obj.reg.ctrl

USIDelegate

(obj.reg.ctrl)

Query sc_sim_context for 'obj.reg.ctrl'

Try to convert to registered interfaces

ctrl = usi.find ('obj.*.ctrl')
Initialization

sc_register

constructed
obj.reg.ctrl

Elaboration

Python

USIDelegate

ctrl = usi.find
('obj.*.ctrl')

UsIDelegate
('obj.reg.ctrl')

Query sc_sim_context for 'obj.reg.ctrl'

Try to convert to registered interfaces

dynamic_cast to
sc_object

sc_simulation_context
```python
ctrl = usi.find ('obj.*.ctrl')

USIDelegate ('obj.reg.ctrl')

Query sc_sim_context for 'obj.reg.ctrl'

Try to convert to registered interfaces

dynamic_cast to sc_object

Create

sc_simulation_context

constructed obj.reg.ctrl
```
Initialization

Elaboration

1. `ctrl = usi.find ('obj.*.ctrl')`
2. `USIDelegate (`obj.reg.ctrl`)`
3. Query `sc_sim_context` for `obj.reg.ctrl`
4. Try to convert to registered interfaces
5. `dynamic_cast to sc_object`
6. Lookup on `scireg db` for `obj.reg.ctrl`
7. Create

Scripting environment
- Python
- USIDelegate
- `sc_register`

SystemC
- `sc_object`

Plug-in (Util) APIs
- `scireg`
- AHBDevice

Create
ctrl = usi.find('obj.*.ctrl')

USIDelegate ('obj.reg.ctrl')

Query sc_sim_context for 'obj.reg.ctrl'

Try to convert to registered interfaces

dynamic_cast to sc_object

Create

Lookup on scireg db for 'obj.reg.ctrl'

dynamic_cast to AHBDevice not possible

Create

constructed obj.reg.ctrl

sc_simulation_context

Python

USIDelegate

sc_register

Python

sc_object

C++

PY

scireg

C++

PY

AHBDevice

C++

PY

Scripting environment

SystemC

Plug-in (Util) APIs

© Accellera Systems Initiative
ctrl = usi.find ('obj.*.ctrl')

USIDelegate ('obj.reg.ctrl')

Query sc_sim_context for 'obj.reg.ctrl'

Try to convert to registered interfaces

constructed obj.reg.ctrl

sc_simulation_context

dynamic_cast to sc_object

Create

Lookup on scireg db for 'obj.reg.ctrl'

Create

dynamic_cast to AHBDevice

not possible

Create

None
ctrl = usi.find ('obj.*.ctrl')

USIDelegate ('obj.reg.ctrl')

Query sc_sim_context for 'obj.reg.ctrl'

Try to convert to registered interfaces

dynamic_cast to sc_object

Create

Look up on scireg db for 'obj.reg.ctrl'

dynamic_cast to AHBDevice not possible

Create

None

tuple of implemented interface proxies (SWIG)
Plugin function call
Or how to access simulation information via interface delegation
Plugin function call
Or how to access simulation information via interface delegation

Scripting environment
Python
USIDelegate
sc_register
ctrl.
basename()

SystemC

Plug-in (Util) APIs
sc_object
C++
PY

Simulation

© Accellera Systems Initiative
Plugin function call

Or how to access simulation information via interface delegation

- Scripting environment
- SystemC
- Plug-in (Util) APIs

Python
USIDelegate
sc_register
C++
PY

ctrl. basename()

__getattr__()
Plugin function call
Or how to access simulation information via interface delegation

- Scripting environment: Python
- SystemC
- USIDelegate
- sc_register
- sc_object
- C++
- Plug-in (Util) APIs
- PY

ctrl.basename()

__getattr__()

Query implemented interface proxies (SWIG) for basename()
Plugin function call

Or how to access simulation information via interface delegation
Plugin function call

Or how to access simulation information via interface delegation

Scripting environment
  Python
  USIDelegate

SystemC
  sc_register

Plug-in (Util) APIs
  sc_object

C++
  PY

Simulation

ctrl. basename()

__getattr__()

Query implemented interface proxies (SWIG) for basename()

Call basename()

basename()

SWIG Proxy
Plugin function call

Or how to access simulation information via interface delegation

Scripting environment

Python

USIDelegate

sc_register

SystemC

Plug-in (Util) APIs

sc_object

C++

PY

ctrl.
basename()

__getattr__() Query implemented interface proxies (SWIG) for basename()

Call basename()

basename() SWIG Proxy

Call C++ Function

Simulation
Plugin function call

Or how to access simulation information via interface delegation

Scripting environment
Python
USIDelegate
ctrl.
basename()
__getattr__()
Query implemented
interface proxies
(SWIG) for basename()
Call basename()

SystemC
sc_register

Plug-in (Util) APIs
sc_object
C++
PY

Call C++
Function
basename()
SWIG Proxy
virtual
basename()
Plugin function call
Or how to access simulation information via interface delegation

Simulation

Scripting environment
Python
USIDelegate
ctrl.
basename()

SystemC
sc_register
__getattr__()
Query implemented interface proxies (SWIG) for basename()
Call basename()

Plug-in (Util) APIs
sc_object
C++
PY

basename() SWIG Proxy

Call C++ Function

virtual basename()

basename()
Simulator agnostic

✓ Accellera SystemC

✓ Mentor Graphics Questasim
  – No sc_main
  – Introduction of a help macro
  – USI_MODEL_EXPORT
  – GCC 32 bit
  – Debugging problems
  – Potential for deeper integration
Language agnostic

✓ TCL
✓ Ruby
✓ Python
  ◦ Lua
API agnostic

✓ sc_object, tlm_socket

✓ GreenControl gs_param

✓ Cadence scireg

✓ AHB/APBDevice
Usability vs. Speed

• Easier to write
• Faster prototyping
• Domain Specific
  – Statistics
  – Text processing

• Slower (Python: 10 to 100 times)
• Fast extensions available (e.g. Python pandas)
• Ideal for pre-/post processing and glue
SoCRocket is available online: https://socrocket.github.io/

For more information please contact us!
Questions