SST-SystemC Interoperability Toolkit (sstscit)

Sabbir Ahmed

April 4, 2019

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

1 Introduction

A toolkit to provide interoperability between Structural Simulation Toolkit (SST 8) (with SST Elements 8) and SystemC 2.3 (with Transaction-Level Modeling Library (TLM 2)).

This collection of header files provides methods to transmit and receive signals between SST components and SystemC modules. The toolkit provides a black box interface that can be interfaced with both SST and SystemC via their internal communication transports.

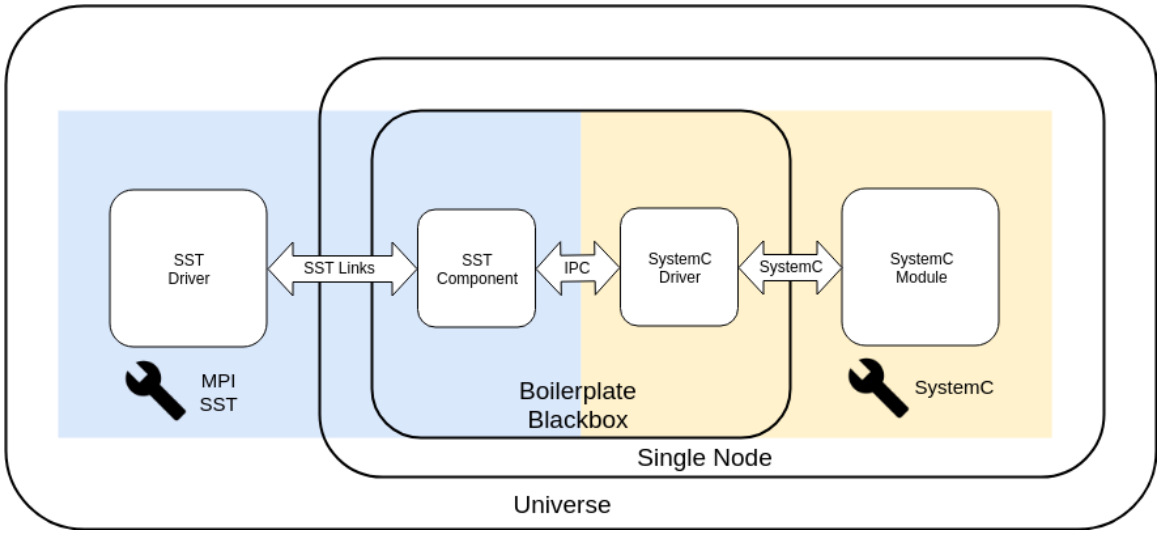

Figure 1: Components of sstscit

2 Components

2.1 Black Box Interface

The black box interface consists of:

1. A SystemC driver
2. An SST component

Each SystemC modules must have their corresponding driver file to interoperate within the black box interface. It is possible to interoperate multiple SystemC modules with a single driver file. However, the additional communication lines must be accounted for in the corresponding black box SST component.

2.2 Headers

3 Communication

3.1 Inter-Black Box Communication

The two components inside the black box interface are spawned in the same node and therefore communicate via interprocess communication (IPC) transports. The following is a list of supported IPC transports:

1. Unix domain sockets
2. ZeroMQ

It is possible to add custom IPC protocols to the interface.

3.2 SST-Black Box Communication

An SST model can interface the black box via standard SST links.

The following snippets demonstrate an SST link transmitting a unidirectional signal from the SST environment to the black box interface.

```
// parent_sst.cpp

// register a string event in the class declaration
SST_ELI_DOCUMENT_PORTS(
    { "demo_din", "Demo model data in", { "sst.Interfaces.StringEvent" }},
    ...
)

// initialize the link in the class declaration
SST::Link *demo_din;

// set up the SST link in the constructor
demo_din = configureLink("demo_din");

// trigger the event in the clocked function
demo_din->send(new SST::Interfaces::StringEvent(...));
```

```
// blackboxes/demo.cpp

// register the same string event in the class declaration
SST_ELI_DOCUMENT_PORTS(
    { "demo_din", "Demo model data in", { "sst.Interfaces.StringEvent" }},
    ...
)

// initialize the same link in the class declaration
SST::Link *demo_din;
```

```
// set up the SST link in the constructor with an event handler
demo_din = configureLink(
    "demo_din",
    new SST::Event::Handler<demo>(this, &demo::handle_event)
);

// receive and parse the event in the event handler
void demo::handle_event(SST::Event *ev) {
    auto *se = dynamic_cast<SST::Interfaces::StringEvent *>(ev);
    if (se) {
        std::string _data_in = se->getString();
        ...
    }
    delete ev;
}
```

3.3 SystemC-Black Box Communication

A SystemC module can be interfaced by a standard source file inclusion.