The modeling and simulation environment for proprioceptive systems

Short Description and User Guide

A. The main properties of the environment:

- Based on the defined reference architecture for proprioceptive nodes presented in the following figure.

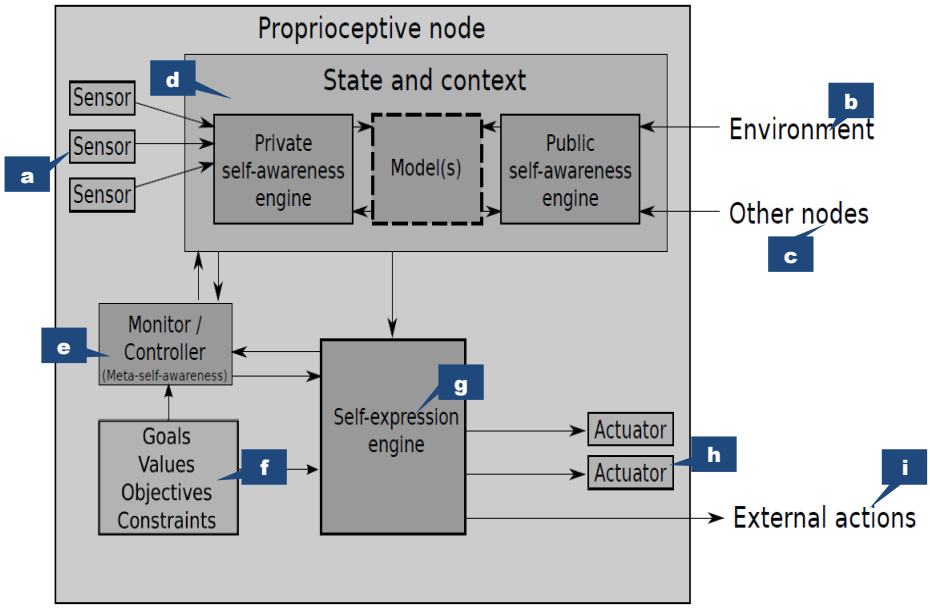

Figure A.1: Reference Architectural Framework of a proprioceptive node

- Transaction-Level Modeling:
  - Focus on the communication details between components of a system
  - Separation between communication and computation. The latter is added by the user according to the system under investigation.

- Loosely Timed Modeling Style
  - Temporal Decoupling of process.
  - Keywords: quantum keeper, Blocking transport method

B. It is composed of the following elements (cf. figure B.1):

- TLM components:
  - SenEnv* (a, b)
  - OtherNode* (c)
  - IC1, IC2, IC3, IC3
  - SAE (d)
  - GVOC (f)
  - Monitor (e)
  - SEE (g)
  - Actuator* (h)
  - ExtActions* (i)

  Except for the IC1-IC4 whose function is to route transactions and, when necessary, to perform address translation, each of these components embodies or is a part of a conceptual component of the defined reference Architecture for a self-aware and self-expressive node. The corresponding component of the architecture is indicated by the letter(s) within the brackets.

  * There can be more than one of these components in a node.

- A SystemC Module
  - Node

  This is the parent module of the component cited above. It is responsible for the node construction during elaboration, in details for the component generation and instantiation, sockets and port binding within the nodes, for the

- C++ classes
  - ActCore
  - LmCore
  - SeeCore

  These are respectively the computation core of the Actuator, LModel and the SEE TLM-Components. These classes should be used to implement the functionality of the concerned components for more clarity as well as to avoid inflecting changes in the threads' implementation, which could lead to an undesired behavior of the model or to wrong simulation results, especially if one's not confident with the SystemC simulation procedures.

- FIFO Channels, in- (SenEnv, OtherNode) and output ports (ExtAction)
  - For the communication of a node with the outside world.

Port-to-FIFO/FIFO-to-Port bindings within the nodes are performed automatically during elaboration in the module node, as already mentioned.

Port-To-FIFO/FIFO-to-Port bindings among nodes must be performed manually by the user.

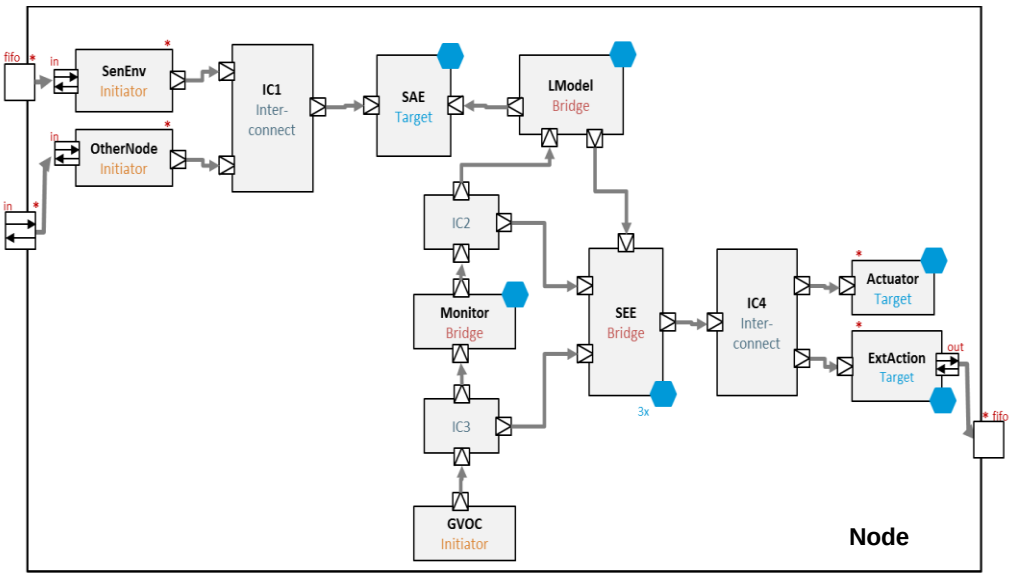

Figure B.1: Component View of the Environment

C. Building a simulator

To be able to simply use this environment and develop a simulator for a concrete system, we recommend you the following steps:

1. Clearly defined the structure of your system:
    a. Number of nodes
    b. Topology
    c. The number of actuators, sensors within each node

2. Clearly defined the functionality of the respective nodes' components and implement it in separate files and/or as classes. Do not forget to include case distinctions to lead the components of a node to the functions specially intended for them.

3. Determine the data types and implement appropriate conversion functions
    a. Data are exchanged or rather passed between TLM components exclusively as pointer of type *unsigned char* through the transaction object.

4. Instantiate the nodes with respect to the given limits for the respective constructor variables and connect them (bindings operations) in accordance with your system topology (cf. 1.)

5. Start the simulation

D. More:

- There is a file named "S*ystemC.props*", which shows an example of the "must-do" project configurations to be able to use to run a systemC program in Microsoft Visual Studio, provided SystemC has already been installed.

- There is also a file named "*Doxyfile*" which shows the "should-do" Doxygen configurations in order to create a documentation adapted to SystemC Projects using Graphiz for the graphs.

- There is a file named "*SimEnv_Description"* which contains a detailed description of the environment, the processes, the execution chronology of processes and transactions and the key parameters of the model.