

Uploading and Running Configuration Application on Sensor Nodes

This demo simulates a scenario where a software component is uploaded to sensors and configures them afterwards. It can be considered as a starting point for developing a healthcare application in the Vitruvius project. For example, a doctor works with a system to configure patients' body sensors for checking their medical status. The configuration contains defining services which include the conditions upon which sensors measure physiological events and handle the occurrence of respective events. The sensor configuration is formatted into an executable code, execution of which results into configuration of the sensor nodes in the network.

The software toolchain used in the demo consists of three main programs [1, 2] namely *simulator*, *compiler* and *loader*. Figure 1 depicts the working flow.

Step 1: Running simulator by using a network layout description (.wnl) which specifies the location and the type of the nodes in the network to the simulator.

Step 2: The simulator generates a symbol table and *Node simulator* GUI which displays the simulated nodes. There are three types of simulated nodes: *sensor nodes* with IDs 17, 19, 32 and 65, *gateway node* and *backend node* with IDs 50 and 68 respectively. In the symbol table, the names of system calls and handlers are mapped into series of IDs so that the communicated messages remain small.

Step 3 and 4: Using the information in the symbol table, the compiler compiles a .wsp file which defines services and their subscriptions. For example, the gateway node subscribes to the sensor nodes and the backend node subscribes to the gateway node. A *bytecode* (.wbc) file containing configuration messages is generated. Configuration messages are virtual machine instructions which execute service deployment on nodes and events subscription.

Step 5 and 6: The loader takes the configuration messages and broadcasts them to simulated nodes (resided in *Node simulator* GUI).

Step 7: The simulated nodes receive the messages. Based on Content Based Addressing [1], services are deployed, configured and started on corresponding nodes. A run of the network is displayed afterwards.

A running example is that a sensor node is configured to insert some randomly generated values to its local buffer and later sends them to the gateway node. The gateway node aggregates received values from all the sensor nodes and forwards them to the backend node. The backend node writes these values into a file. In *Node Simulator* GUI, received values are indicated as *nodeID : value*. Similarly, writing these values into the file has been indicates as "*writing sensor value*" + *value*.

References:

1. R. Bosman, J. Lukkien, R. Verhoeven, "An Integral Approach to Programming Sensor Networks", CCNC 2009.
2. <http://www.win.tue.nl/~rbosman/toolchain.html>

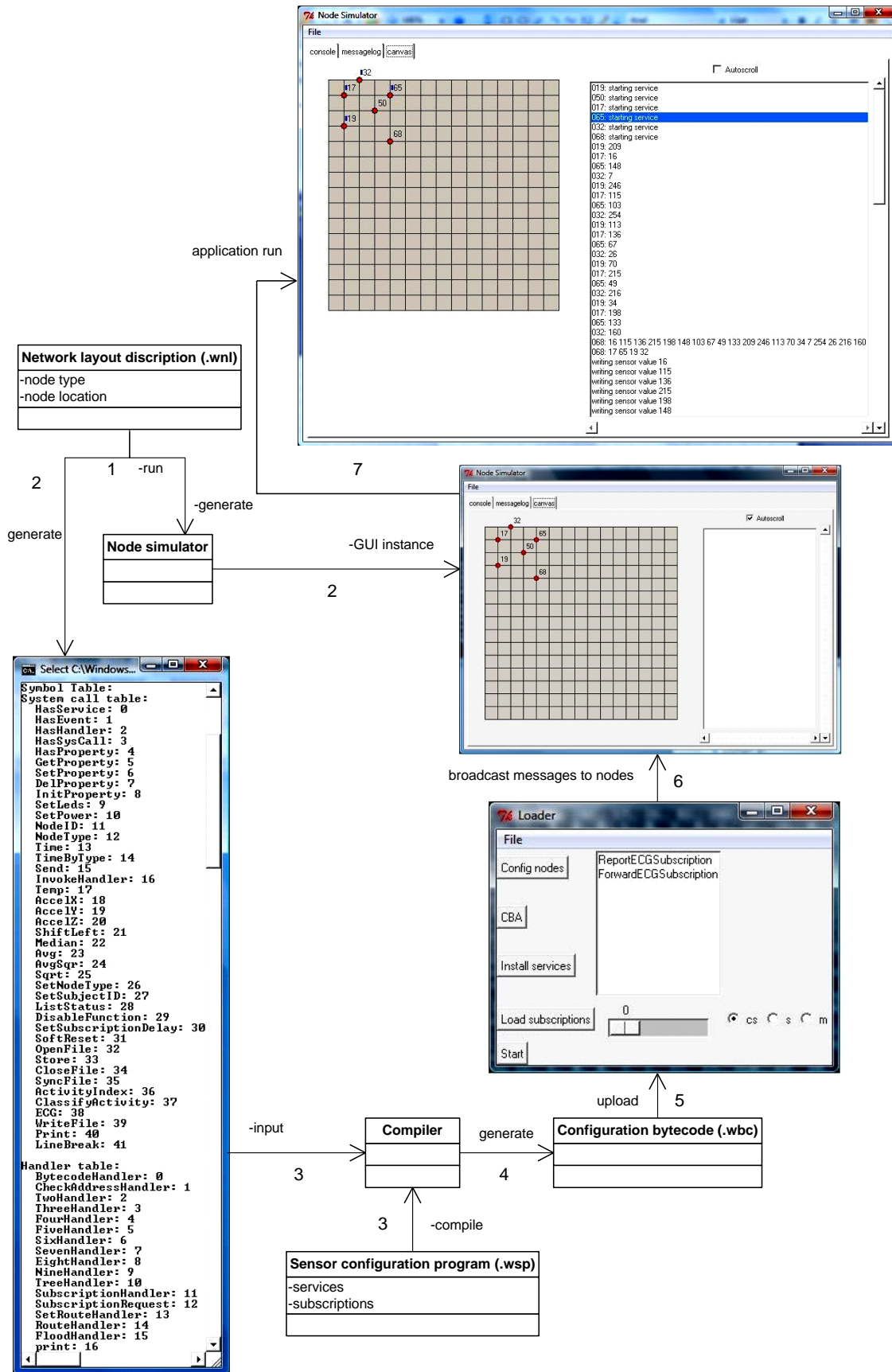


Figure 1: Working flow for software toolchain