Prototype Review Document

Frank Clark
francis.j.clark@drexel.edu

Dustin Ingram
dustin.s.ingram@drexel.edu

Maria Kolakowska
maria.j.kolakowska@drexel.edu

Aaron Rosenfeld
aaron.rosenfeld@drexel.edu

November 25, 2010

1 Introduction

The prototye represents the integration of a simulator with basic functionality as well as a Display Tool (SDT). The prototype fulfills a number of goals, as well as frames the work needed to be done in the next version.

1.1 Prototype Goals Completed

- Created a simulator which aggregates a user-defined Scenario and Network.
- Specified a rough format for the user-definition of Scenarios, Networks and Agents.
- Implemented a Display Tool based on a real-world model which displays, verbatim, the real-time events created by the simulator.
- Allowed agents to react in real-time to events distributed through the simulator by other agents.
- Created a simplified process for using an Agent to publish real-world data into the simulator.
- Created an API channel for the simulator which allows the subscription and publication of events.

2 Sample Configuration Files

The following files represent the sample configuration files for the Scenario Definition and the Network Topology Definition, which are interpreted by the simulator to set up the virtual nodes, their agents, and settings.

2.1 Scenario

This sample Scenario Configuration defines a scenario with 5 nodes, each with a local 'Moving' agent which has been independently defined to take a random walk within the limits of the scenario. The configuration makes the node model, applies the user-defined agent, and adds the node to the Scenario.

```
from stage.scenario import Scenario
from stage.model import Model
from stage.agents.testagent import TestAgent
from stage.agents.randommoveagent import MovingAgent
from stage.agents.circlewalker import CircleWalkerAgent
```

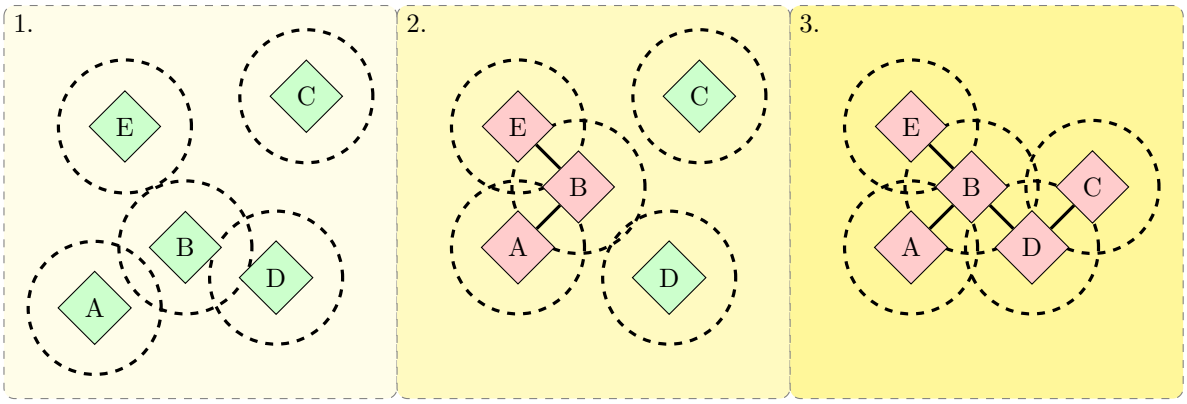


Figure 1: A diagram of the Prototype: *1.* Agents have been configured to perform random walks within a boundary, and each contains a wireless interface. *2.* When nodes recieve an event indicating they are in range of another node, they hold their position and create a network link. *3.* A static network is formed.

```
class ScenarioDef(Scenario) :
    def __init__(self) :
        Scenario.__init__(self)

        for node_id in range(0, 5) :
            node = Model(name='n%s' % node_id, position=(0,0,0), \
                agents = [], interfaces = {})
            agent_instance = MovingAgent()
            node.get('agents').append(agent_instance)
            self._nodes.append(node)
```

2.2 Network

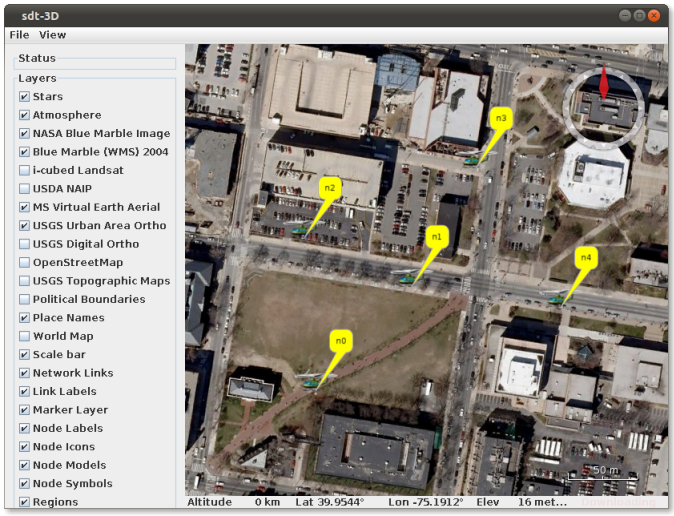
This sample Network Configuration sets one wireles interface for each node in the scenario.

```
from stage.network import Network
from stage.model import Model

class NetworkConcrete(Network) :
    def __init__(self, scen_inst) :
        Network.__init__(self, scen_inst)

        for node in self._scen_inst.get_nodes() :
            node.get('interfaces')['eth0'] = Model(type='wireless', \
                range=20, ssid='wlan0')
```

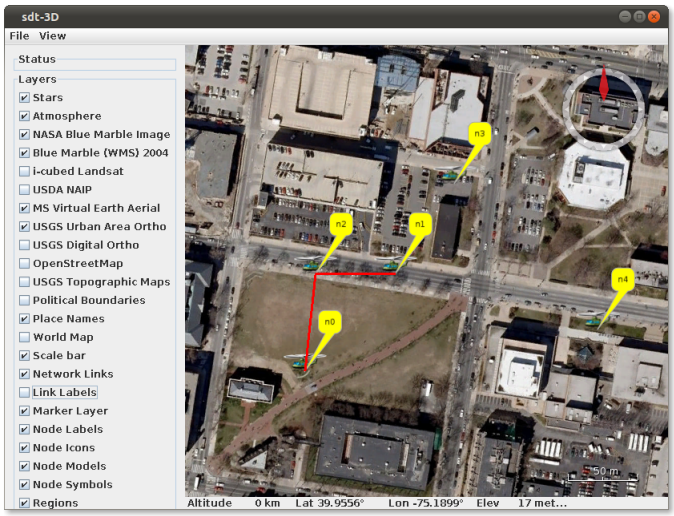
3 Diagram of Prototype Scenario

4 Screenschots of SDT-3D (STAGE Display Tool)

The nodes in their initial state:



After a few ticks, some nodes have become linked and can communicate:



All nodes are now connected to form a static network: