Distributed Computing User Interfaces for 3Worlds Ecological Simulation Framework

Name: Weiwei Hou
ID: U5202546
Supervisor: Dr. Shayne Flint, Dr. Ramesh S
Outline

• Introduction
  – Distributed Computing
  – 3Worlds
  – Project tasks (deployment manager widget, playback simulator)

• Objectives

• Implementation
  – Two interfaces of Deployment manager widget
  – Playback, state machine, playback controller widget

• Test

• Conclusion
Introduction

• Distributed Computing
  – Deal with large computational problem by divided it into several tasks, and solved by multiple computers

• 3Worlds
  – Distributed Ecological Simulation Framework

• Tasks
  – Deployment manager widget
  – Playback simulator
Objectives

- Improving the understanding of distributing system.
- Ability to plan and execute the testing of distributed computing systems.
- Ability to develop and test graphical interface components for use within a distributed environment.
- Ability to collaborate with members of an international research group.
Background

• Graph Framework
  – Node, Edge, property

• Node communication
  – Sending and receiving messages

• “Experiment”
  – A distribute computing model

• State Machine
  – State Transaction
Deployment Manager Widget

- Simple Interface
- Advanced Interface
Playback Simulator

- Architecture
Playback Simulator

• Message controller

```java
Find node, which has property of "Recording".
Initialise() {
    Open a new file;
}
CallRendennzvous(messages) {
    Write messages into this file;
}
Finalise() {
    Close the file;
}
```

State Machine

- Logger1
- Logger2
- PlaybackNode
- PlaybackControlWidget

Messages

Recording

State Machine Widget

Playback Simulator
State Machine

• Six State
  – Initializing
  – Stepping
  – Running
  – Pausing
  – Finished
  – Quitting
Time Widget

• Time Setting
  – Set the boundary to read the messages within a certain period of time.

• Delay Setting
  – Insert a number of millisecond to set the delay for sending message from the playback node to the target node
Testing

- Test Recording
  - Test graph
    - One sender node and three receiver nodes
    - Four messages to be sent
  - Results
    - Four messages sent by sender are stored into the repository
Test

• Test playback simulator
  - Graph
    • Target Logger Nodes
    • One playback node
  - Result
    • Target logger nodes receive the messages one by one

```python
logger1
[MessageHeader type=1, sourceNodeId=..., targetNodeId=...]
logger2
[MessageHeader type=1, sourceNodeId=..., targetNodeId=...]
logger3
[MessageHeader type=1, sourceNodeId=..., targetNodeId=...]
logger1
[MessageHeader type=2, sourceNodeId=..., targetNodeId=...]
logger2
[MessageHeader type=2, sourceNodeId=..., targetNodeId=...]
logger3
[MessageHeader type=2, sourceNodeId=..., targetNodeId=...]
```
Test

- Playback controller
  - State Machine
  - Time widget
Conclusion

• Distributed Computing and 3Worlds
• Related parts
  – Graph, “Experiment”, State machine
• Implementation
  • Deployment Widget (Simple, advanced interface)
  • Playback simulator (msg records, simple playback, state machine, controller widget)
• Test
• Future work
  – Implement the deployment manager