Graphical Components for Directed Diffusion Simulator

Abstract

*This document presents the design for a series of graphical components to the Directed Diffusion Simulator developed by the 2012 ECE 7650 Advanced Computer Network Architecture Course. UML techniques are used to present class structures and operational concepts. The components are designed and implemented to permit stand-alone or concurrent use. As/if work continues on the simulator more components should be developed to enhance the overall simulator quality.*

# Introduction

Simulation packages/frameworks typically provide the means for users to see an organized/cohesive view of the data ase the

# Background

Directed Diffusion Simulator provides several classes that are framework to the. The original implementation of the Directed Diffusion Simulation (DDSim) uses console-based output in its examples. Whjen the

# Design Requirements

The general design requirements for DDSim UI components were:

1. Provide a plotting mechanism for the data being collected:
   1. Node energy variance was identified as a requirements.
   2. Provide a means the trace data used for plotting operations.
2. Produce general, extendable, decoupled components t

The design

## Decoupled Modules

The components could not b e

### Data Gathering

The developed components

Strong de-coupling between code modules was a requisite for updating the

# Baseline Modifications

During the initial prototyping, the number of classes in the default package grew to unsustainable levels. As a result, the classes were migrated from the

The implementation:

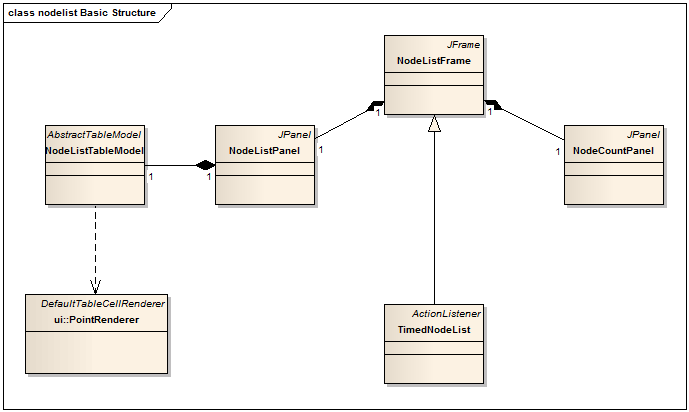
1. dd – the original
2. dd.ui – generic classes used for supporting the
3. dd.ui.log – contains the logger component related code
4. dd.ui.nodelist – contains the documentation
5. dd.ui.test – general tests for some of the components

# Detailed Designs and Implementation

## Node List

The purpose of the node listing packet (dd.ui.nostlist) is to provide a tabular view of all node data at once – a monitor of sorts. The overall component is implemented as a composition of less-complicated components.

The TimedNodeList class imlements the use case that is most usable to the user. updates the model on a fixed interval.



### NodeListTableModel

The class diagram for the NodeListTableModel is shown in [Figure]. Five columns were selected for the table model:

1. Node Id
2. Location
3. Number of Neighbors
4. Residual Energy

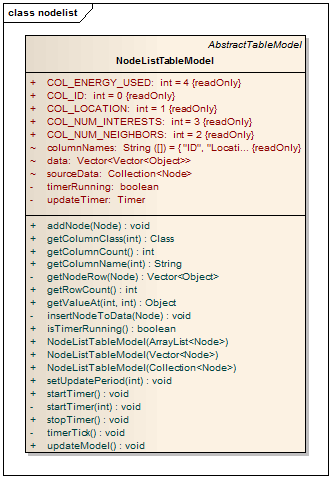


Figure NodeListTableModel Class



NodeLIstPanel

NodeCountPanel

Figure 2 TimedNostList composition

### TimedNodeList

The TimedNodeList class, shown in [reference: figure] implements the functionality most likely usable to the majority

## Log Frame (dd.ui.log)

The dd.ui.log package provides a basic window for posting simulator information into a queue window.

### Supported Logging Levels

As per experiences with other logging systems, the

### 

## Plots

### General

Adding plots to the simulation allows the user to view a trace of information as the simulation progresses. Due to the complexity of implementing a robust and capable plotter from the graphics perspective, a third-party library was selected. After considering a small number of options, the JChart2D library was selected. JChart2D offers a well-documented API, robust examples, flexible output of the image and is straightforward to use, thus making it ideal for the DDSim plotting.

DDSim plotting was designed to allow for static and dynamic data collection. Each collection type is described below.

### Plotting Library

The JChart2D library is an LGPL project that proved most effective in plotting the resource

### dd.ui.plot Hierarchy

The class hierarchy for the dd.ui.plot packet is shown in Figure 3. There are three functional areas to this package:

1. Data Collection: static and dynamic collectors that register lists of nodes to retrieve data for plotting.
2. JFrame derivatives for displaying plots. The PlotFrame classes support static and dynamic data gathering, with an uplink
3. JFileChooser support classes – ImageFilter and derivatives. These classes support the image output of the plots that are generated.

Each of the function areas is discussed below.



Figure 3 dd.ui.plot Package

### Design Details: Plot Data Collection

As per the design constraints

### Design Details: Plotting Frames

## Purposes

## Class Hierarchy

## StaticPlotFrame

The PlotFrame component is an abstract class that comprises of several useful

Using the getTrace() method is not required under ordinary circumstances. The

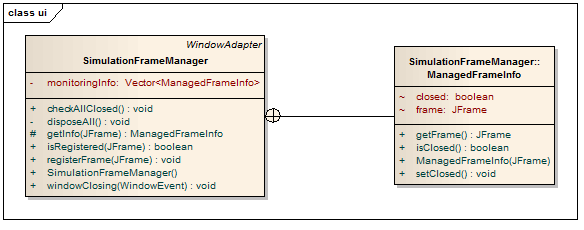
## Results

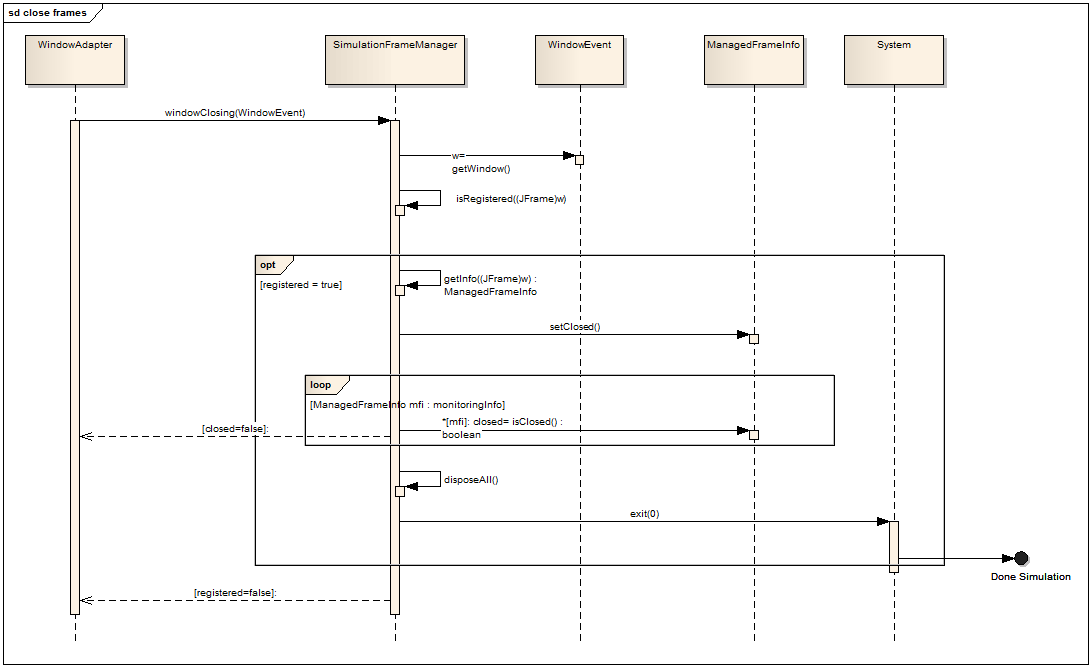
## Window Management

### Design Concept

The design for the frame/window management stemmed from the scenario where multiple windows were opened. In these instances, the JFrame instances cannot set their

The resulting





## Testing Results

The SimulationFrameManager class was successful tested in dd.NodeTestGrapical. Upon receiving the

# Conclusions

The implemented components present