Umeå University
Department of Computing Science

Object-Oriented Programming Methodology 7.5 p
5DV133

OU3 Sensor Network

Submitted
2016-05-04

Authors:
Johan Eklund (kv03jed@cs.umu.se)
Tommie Lindberg (c15tlg@cs.umu.se)
Jakob Lundin (c14jln@cs.umu.se)
Lorenz Gerber (dv15lgr@cs.umu.se, lozger03@student.umu.se)

Instructors:
Anders Broberg
Niklas Fries
Adam Dahlgren
Jonathan Westin
Erik Moström
Alexander Sutherland

Contents

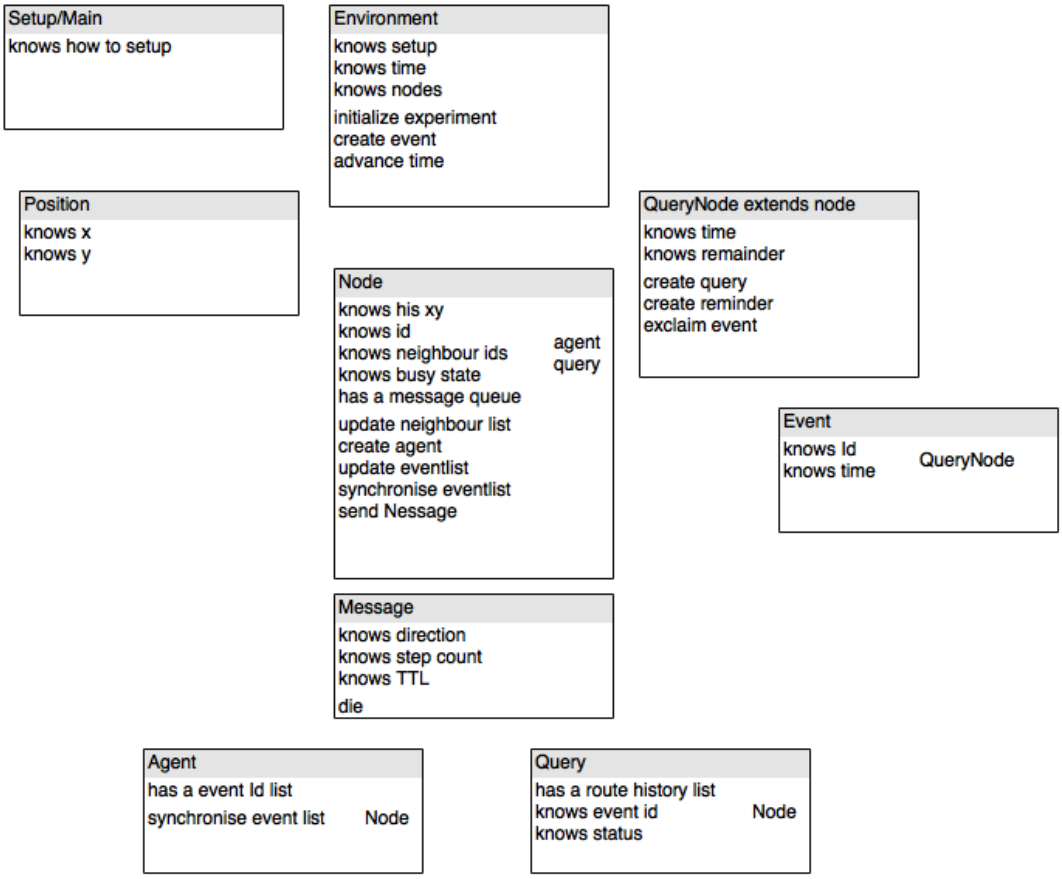

Figure 1: This is the crc diagram

1 Introduction

The assignment was described on the course homepage [4].

1.1 General Design Considerations

In object oriented software design, it is common to build a model representation of the real world system [2] by defining classes the correspond to the real world systems' entities. Here the real world system is a sensor network as described in Braginsky and Estrin [1]. The realworld entities modelled in this assignment can be classified in two main groups: Physical components such as the sensor nodes and non-physical ones, information packages travelling the network, such as the queries and agents. Further, a third type, the environment entity simulates the real surrounding.

Unified Modelling Language (*UML*) and *Class Responsibility Collaborator* (*CRC*) diagrams were composed according to Börstler [2]. The theory of rumour routing is described by Braginsky [1]. Horstman was used as Java language reference [3].

2 Classes Responsibilities and Collaborations

The Class Responsibilities and Collaborations (CRC) diagaram is shown in figure 1.

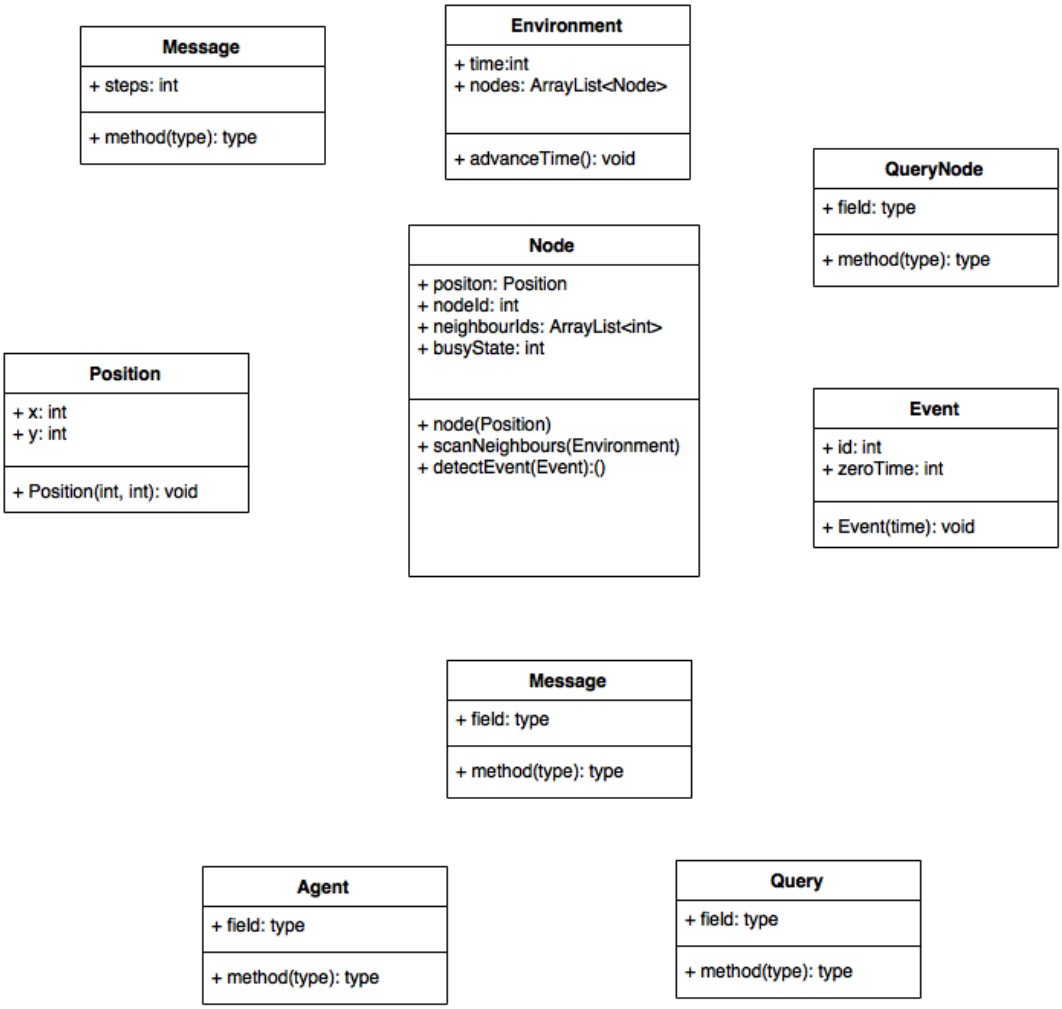

Figure 2: This is the uml diagram

3 Unified Modelling Language Class Diagram

4 Initialization and State Stepping

4.1 Overview

First parameters that are needed to create nodes have to be determined. That includes the position, and the probability to create agents after an event happens. Then nodes can be created and positioned. Afterwards, nodes have to be initialized to know their neighbours. According to Braginsky and Estrin [1], composing the neighbour list could happend either as an initial startup phase, but if nodes are assumed non-static, also continuous. Here nodes were assumed static. Finally, experimental parameters for the environment need to be initialized. These are parameters such as which nodes will create queries and in which frequency. But also the probabilty of events and how the node to experience them is selected.

Setup/Main instantiates the Environment, then instantiates the Nodes with coordinates and places them in the environment. On instantiation, the probability for agent creation is set as attribut in each node.

In each time step, the environment cycles through all nodes. At each node a nubmer of possbile actions are considered and if viable performed. These include: creating an agent,

creating a query, receiving a query, receiving an agent, sending a query, sending an agent.

From the description, it is not clear if creating an agent or a query involves also sending it in the same time step. This behaviour needs to be defined. What happens if a node has one or more messages in the queue, detects an event and is supposed to send an agent. Does the agent end up in the queue? Same question for a query.

First random generation of an event, then to handle the messages (agents and queries). Messages are put in a queue in the node. If a node has either sent, received, or instantiated a new message, a busy flag is set, so that no other operation from outside can be accepted.

5 Testing Framework

References

[1] D. Braginsky and D. Estrin. Rumor routing algorthim for sensor networks. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 22–31. ACM, 2002.

[2] Jürgen Börstler. Objet-oriented analysis and design through scenario role-play. `http://www8.cs.umu.se/kurser/5DV081/CRC_UMINF04.04.pdf`, Umeå University, Department of Computing Science, Umeå, Sweden, 2004. accessed: 2016-04-24.

[3] C. S. Horstman. *BIG JAVA Early Objects*. John Wiley & Sons, New Dehli, 5 edition, 2014.

[4] Umeå University, 5dv133 obligatorisk uppgift 3. `http://www8.cs.umu.se/kurser/5DV133/VT16/uppgifter/ou3/`, 2016. accessed: 2016-04-28.