Water Distribution, Leakage and Quality Control System (WCS)

Second Deliverable of Software Quality Engineering

Aysel Yusubzada (Matriculation Number: 252215)
Karthik Gopanna (Matriculation Number: 255688)
Stefan Ilić (Matriculation Number: 252212)

Table of Contents

1. INTRODUCTION

The second homework consists in deriving a performance model from the UML model of the first homework. It consists of two parts. In the first, we obtain a software model using execution graphs and build a queueing network that represents the hardware platform. In the second part, we describe the AEmilia model.

In addition, we provide the following files:

- MagicDraw reworked UML model:
- JMT models:
    a. Initial model: *WCS.jsimg*
    b. Refactored model: *WCS_refactored.jsimg*
- AEmilia:
    a. Model: *WCS.aem*
    b. Rewards calculation file: *WCS.rew*

2. Modified Sequence Diagrams

Few modifications were made to the Sequence diagrams. The modified sequence diagrams are illustrated in the following figures.

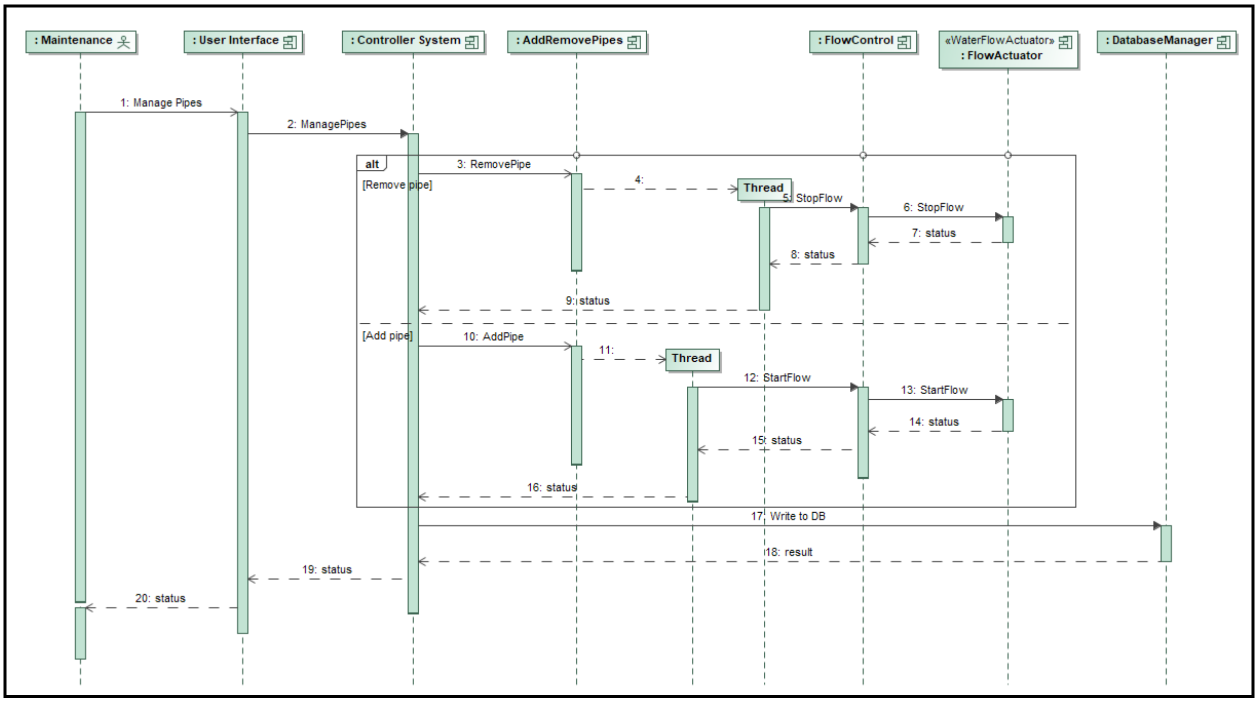

Figure 1. Manage_Pipes Sequence Diagram

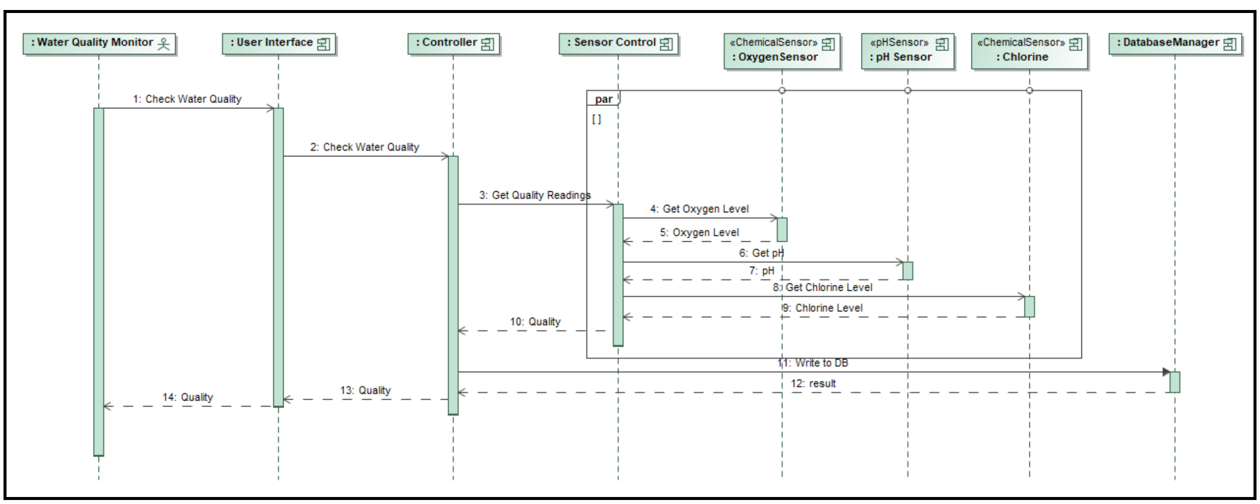

Figure 2. Water_Quality_Monitor Sequence Diagram

3. Annotated Deployment Diagram

The following figure illustrates the Deployment Diagram of our system annotated with CPUs and Disks.

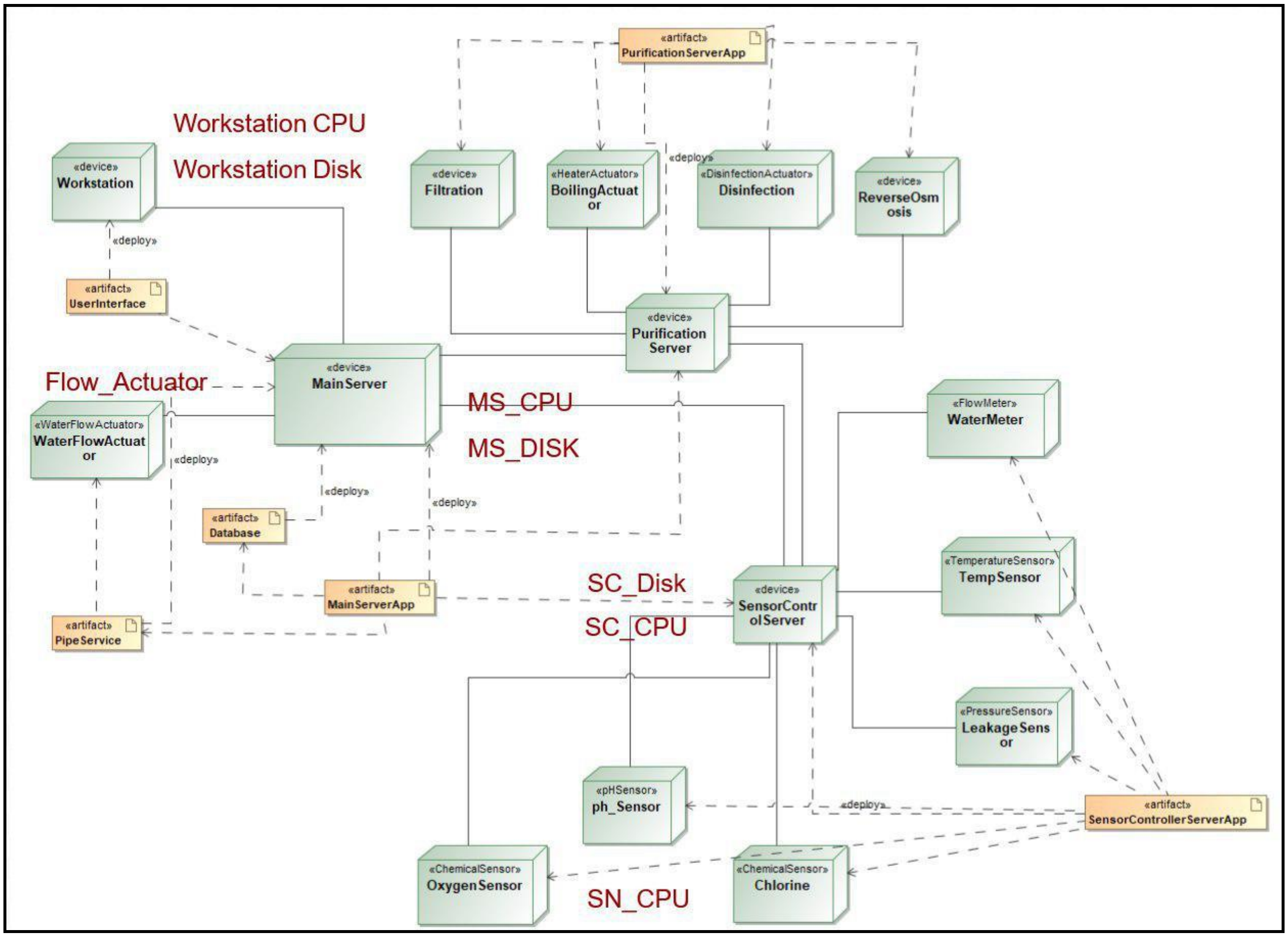

Figure 3. Annotated Deployment Diagram

4. Performance Requirements

We assume following frequency of requests for the two classes of jobs:

- **Manage Pipes:** 1 every 10 seconds.
- **Water Quality Monitor:** 1 every 4 seconds.

The two performance requirements are:

- **System Response Time:** The average system response time for all classes of jobs must be less than 10 seconds.
- **Utilization:** The average utilization of all devices should be less than 70%.

5. Execution Graphs

Two use cases (and therefore two sequence diagrams) are modeled into execution graphs, which are then parameterized. As a result, we obtain the demand vectors for each of the two classes of jobs: *Manage Pipes* requests and *Water Quality Monitor* requests. This software model is then mapped to hardware resources from which average service times are computed.

Software Resources

The basic nodes of the execution graphs are parametrized with software demand vectors (number of visits). The software resources are the following:

1. **WorkUnit**: number of computational operations.
2. **DB**: Number of Database requests.
3. **Net**: Number of messages sent/received over the network.
4. **Sensing**: Number of sensor usages.

The virtual demand vectors are thus vectors of length 4. For example, a basic node requesting 3 work units and 2 database messages will be parameterized with:

$$d = [3, 2, 0, 0]$$

Manage Pipes

Figure 2 illustrates the Execution Graph of the Water Quality Monitor requests.
From the annotated execution graph, we can compute the demand vectors associated with each deployment node by applying the synthesis rules.

$d_{EG1,\ Workstation} = d_1 + d_{18} = [2\ 0\ 1\ 0]$

$d_{EG1,\ MainServer} = d_2 + d_3 + d_4 + d_6 + d_7 + d_{14} + d_{15} + d_{16} + d_{17} = [7\ 2\ 2\ 0]$

$d_{EG1,\ FlowActuator} = d_5 = [1\ 0\ 1\ 0]$

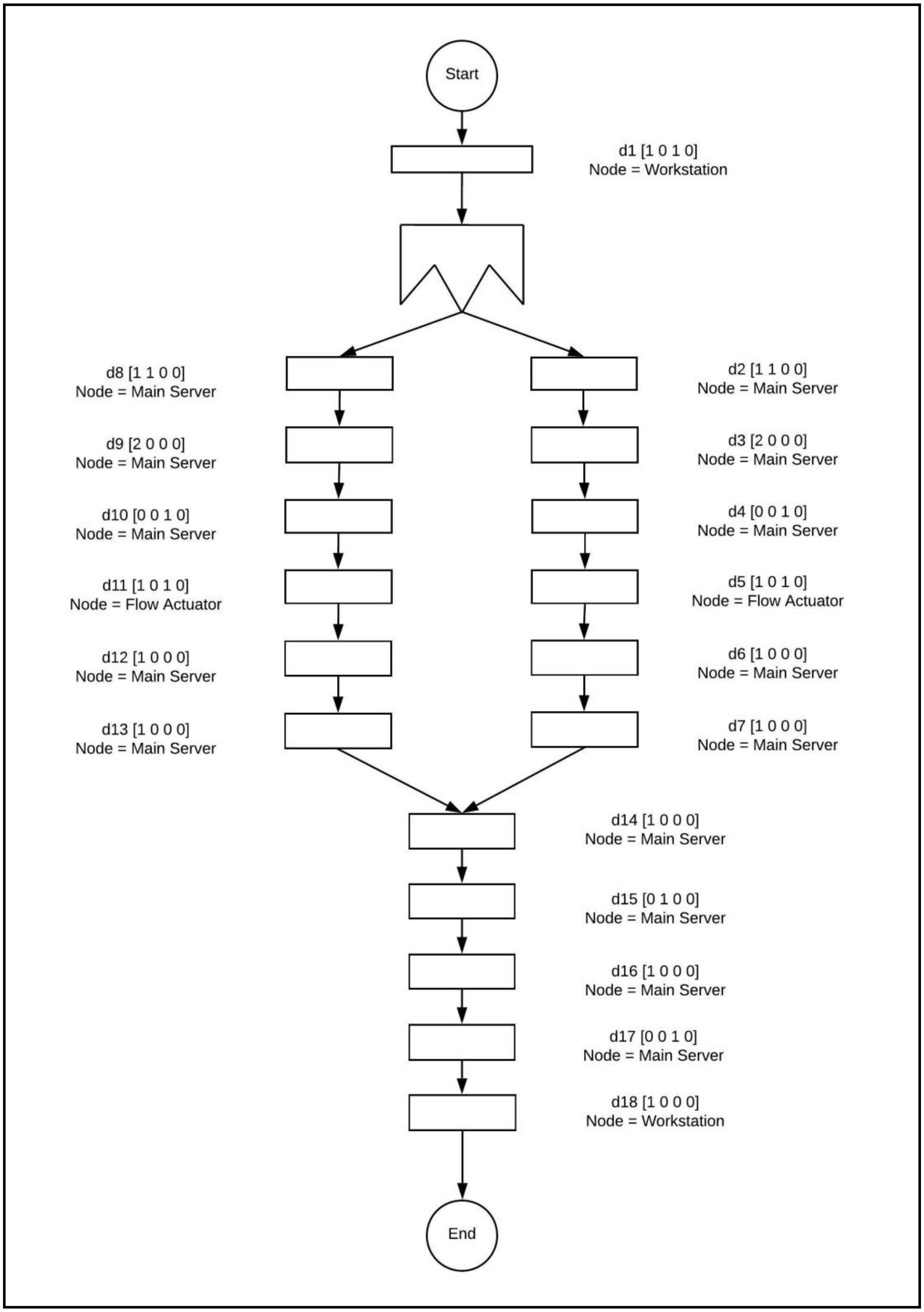

Figure 4. Manage Pipes Execution Graph

Water Quality Monitor

Figure 3 illustrates the Execution Graph of the Water Quality Monitor requests.

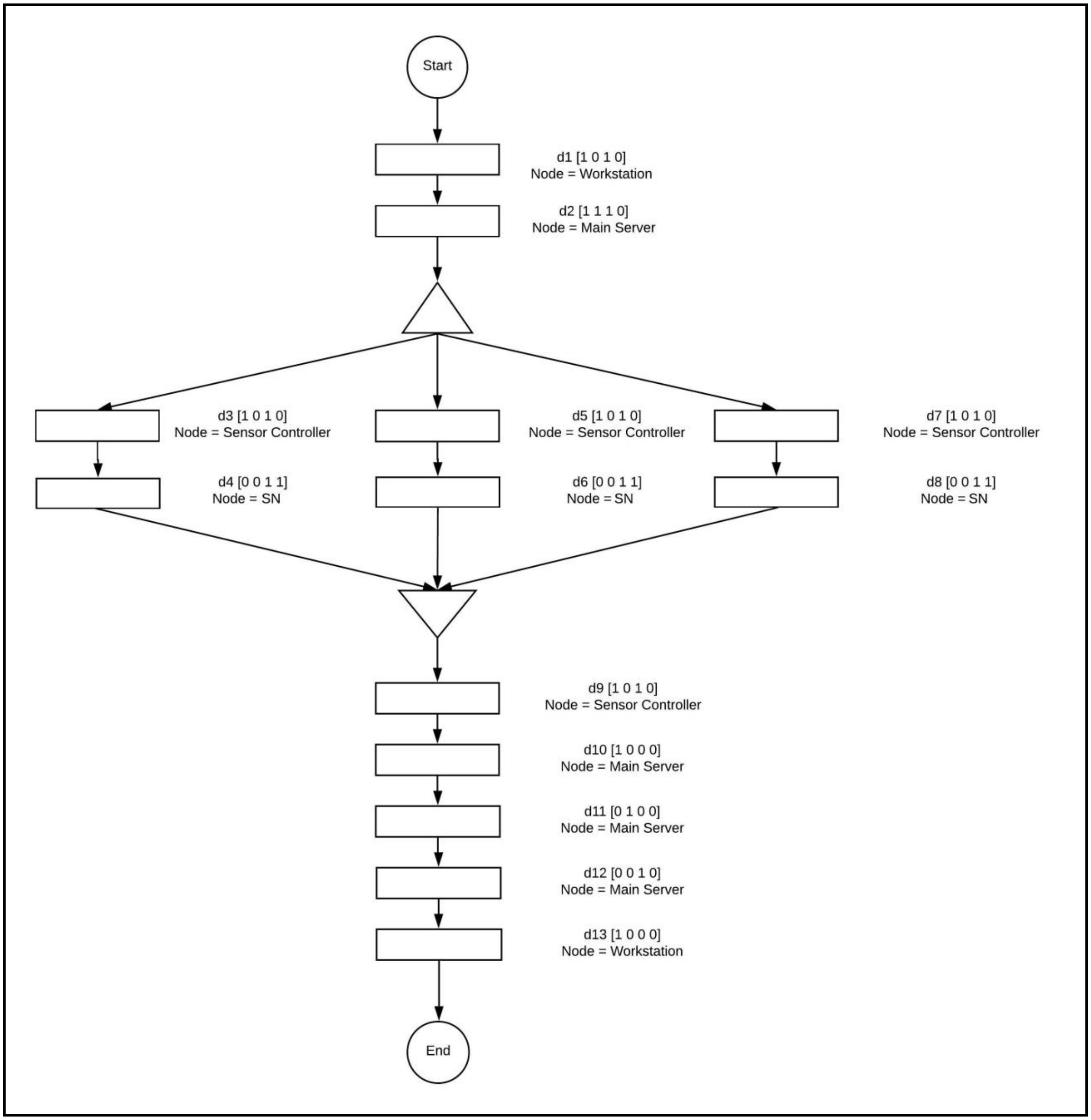

Figure 5. Water Quality Monitor Execution Graph

We now apply the same method to derive the remaining parameterized EGs.

$d_{EG2,\ Workstation} = d_1 + d_{13} = [2\ 0\ 1\ 0]$

$d_{EG2,\ SensorControl} = d_3 + d_5 + d_7 + d_9 = [4\ 0\ 4\ 0]$

$d_{EG2,\ SensorNetwork} = d_4 + d_6 + d_8 = [0\ 0\ 3\ 3]$

$d_{EG2,\ MainServer} = d_2 + d_{10} + d_{11} + d_{12} = [2\ 2\ 2\ 0]$

Overhead Matrix

This step involves obtaining the real demand vectors by means of an overhead matrix. The virtual resources will be mapped to the physical resources of each deployment node, at a lower level of abstraction. Since we have 4 virtual resources and 3 different devices (CPU, Disk, Network), the overhead matrix is a 4x3 matrix.

	CPU	Disk	Network
Work Unit	1300	250	0
Database	200	700	0
Network	100	150	550
Sensing	500	0	0

Table 1. Overhead Matrix

Each cell corresponds to the physical resources required by a unit of a virtual resource. For example, one Work Unit corresponds to 1300 CPU instructions.

The demand vectors are finally obtained by multiplying the virtual demand vectors by the overhead matrix in Table 1.

$d_{EG1,\ Workstation}$ * OM = [2700, 650, 550]

$d_{EG1,\ MainServer}$ * OM = [9700, 3450, 1100]

$d_{EG1,\ FlowActuator}$ * OM = [140, 400, 550]

$d_{EG2,\ Workstation}$ * OM = [2700, 650, 550]

$d_{EG2,\ SensorControl}$ * OM = [5600, 1600, 2200]

$d_{EG2,\ SensorNetwork}$ * OM = [1800, 450, 1650]

$d_{EG2,\ MainServer}$ * OM = [3200, 2200, 1100]

The results are summarized in Table 2.

	WS_CPU	WS_Disk	MS_CPU	MS_Disk	FA_CPU	SC_CPU	SC_Disk	SN
Manage Pipes	2700	650	9700	3450	1400	0	0	0
Water Quality Monitor	2700	650	3200	2200	0	5600	1600	1800
Service Time (seconds)	0.0001	0.0008	0.00006	0.0006	0.0005	0.00006	0.0006	0.0006
MP Service Time	0.27	0.52	0.582	2.07	0.7	0	0	0
WQM Service Time	0.27	0.52	0.192	1.32	0	0.336	0.96	1.08
MP_Lambda Values	3.70	1.92	1.71	0.48	1.42	Null	Null	Null
WQM_Lambda Values	3.70	1.92	5.20	0.75	Null	2.97	1.04	0.92

Table 2.Overhead Matrix Processing

Network delays are assumed to be negligible in our system. But the process of sending messages requires some CPU resources. This is why the network virtual resource is still defined.

6. QUEUING NETWORK

This section describes the hardware performance model. First, the queueing network topology is derived from the deployment diagram. Then, it is parametrized with the results obtained in table 2. The lambda values are assigned to the corresponding service centers in JMT.

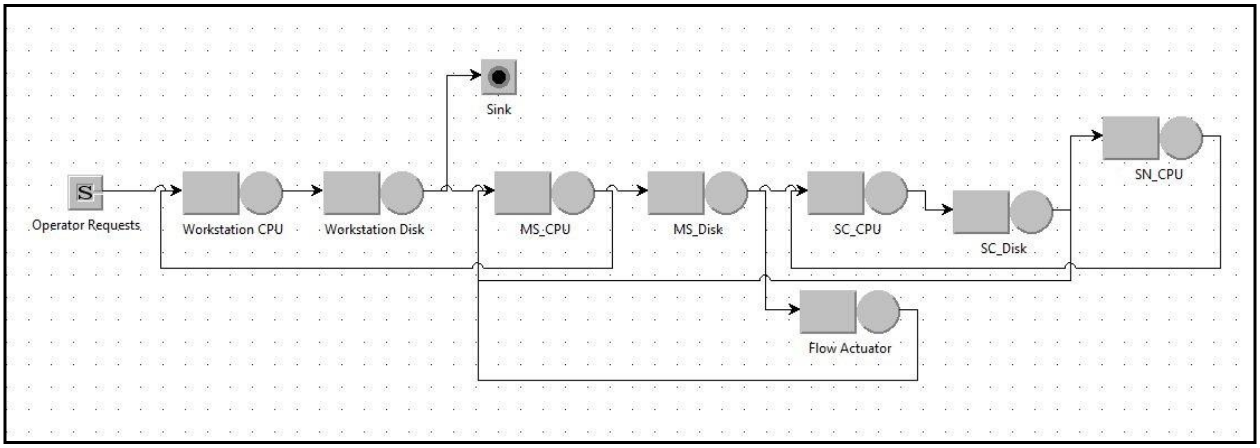

Figure 6. Queueing Network

Classes of Jobs

Two classes of jobs obtained from the two use cases are introduced in the JMT model.
Water Quality Monitor requests
Manage Pipes requests

Simulation Results

The performance indices corresponding to the performance requirements are illustrated in the following images:

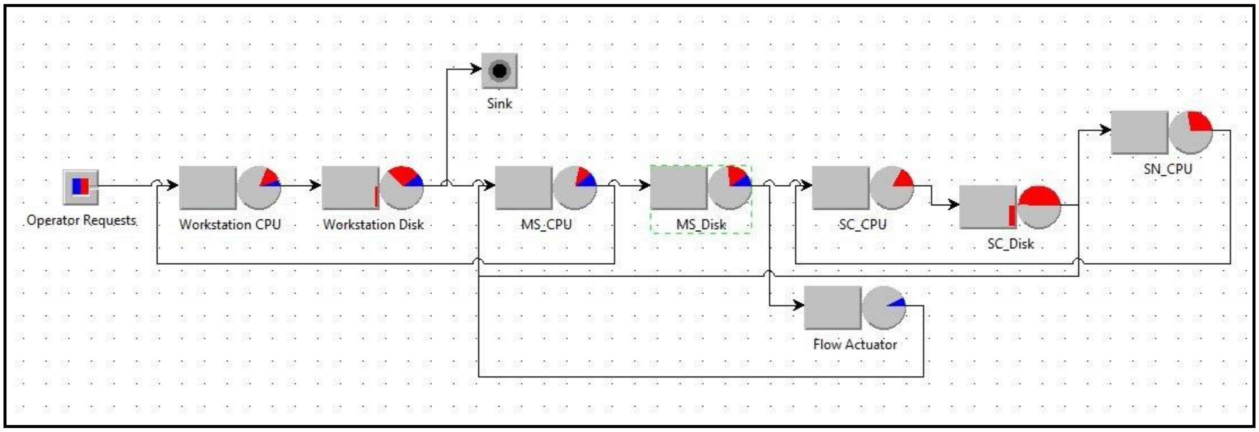

Figure 7. Queue and Utilization

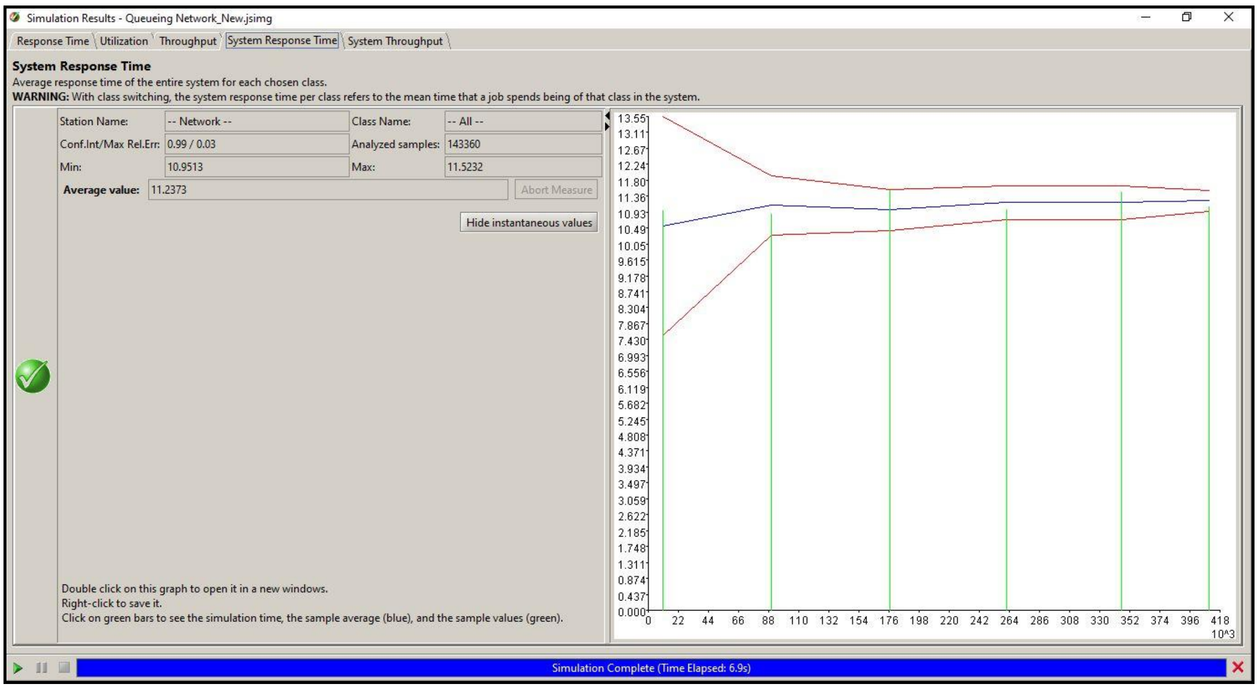

Figure 8. System Response Time

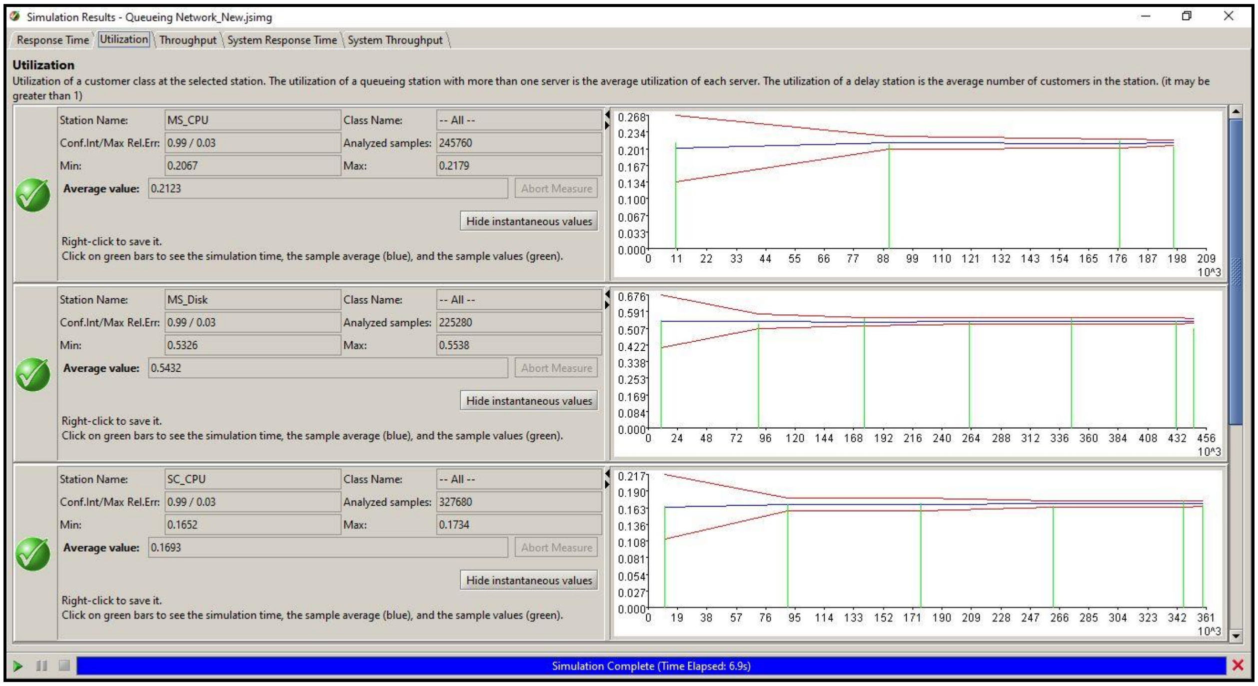

Figure 9. Utilization 1

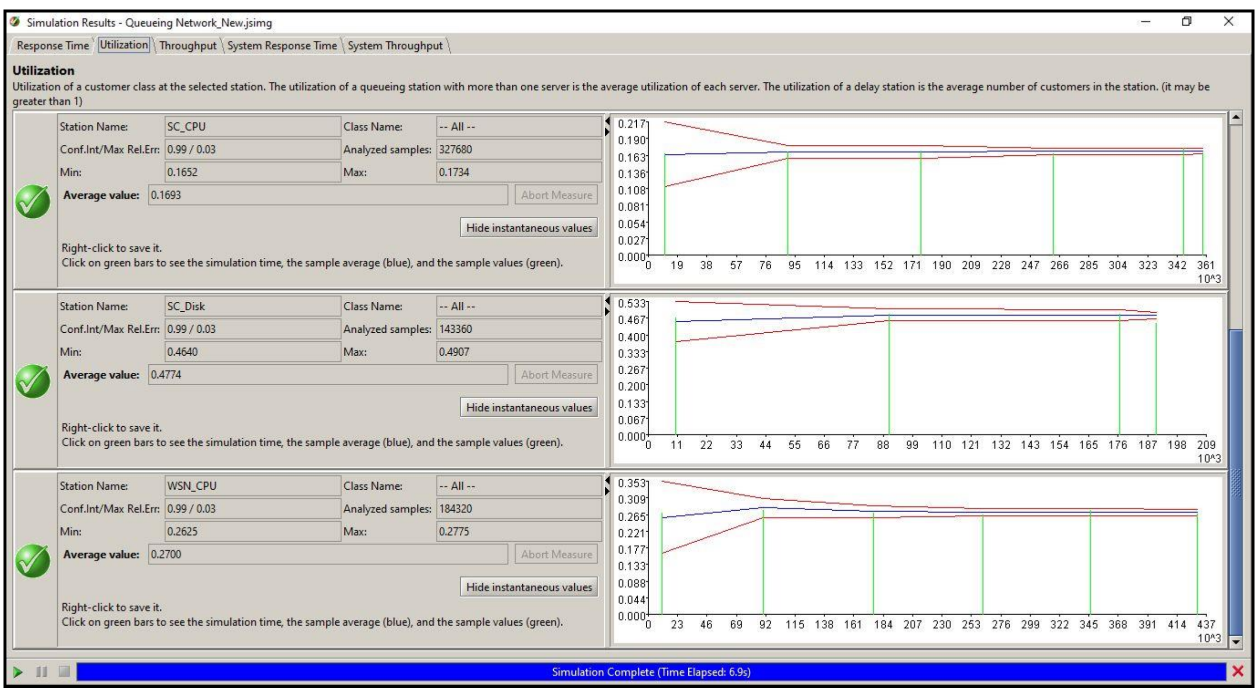

Figure 10. Utilization 2

Bottleneck Identification

It was observed that the utilization of all the devices meets the performance requirements. However, the system response time was observed to be greater than 10 seconds for all classes of jobs. Although the utilization of the MS_Disk service center was observed to be less than 70%, it had the biggest impact on the system response time. To address this problem, a second server was added to the MS_Disk service center.

The refactored Queuing Network illustrated in the following figure. System Response Time requirement is satisfied.

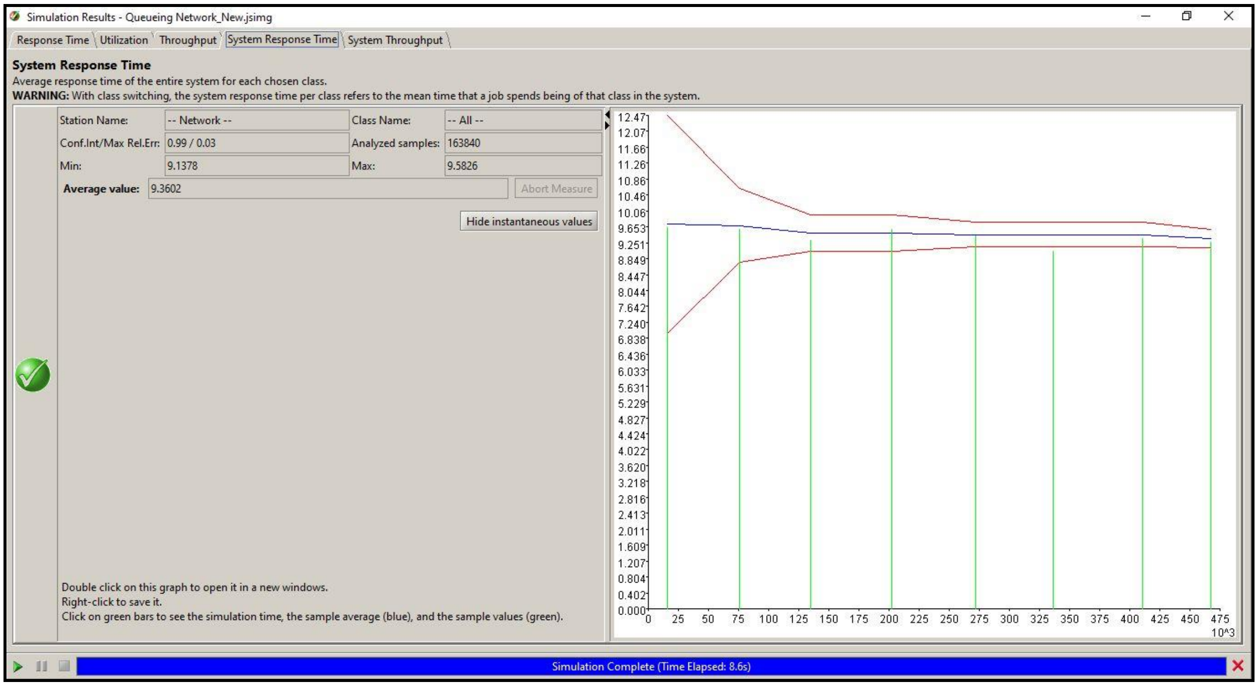

Figure 11. System Response Time after Refactoring

7. AEMILIA MODELING

The AEmilia model can be found in the *WCS.aem* file.

The flow graph of the WCS system is illustrated in the following figure.

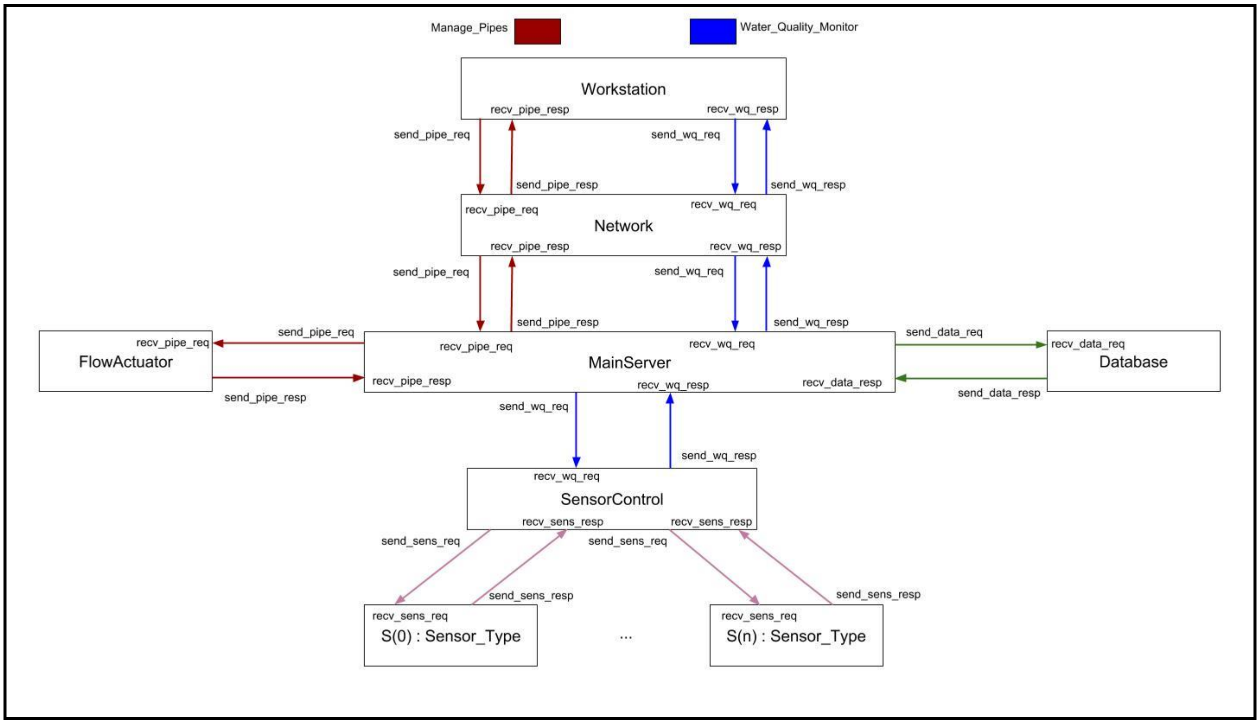

Figure 12. Flow Graph

Our model is composed of the following 7 Architecture Element Types (AET):

1. Workstation
2. Network
3. MainServer
4. FlowActuator
5. SensorControl
6. Sensor_Type
7. Database

Analysis

The performance indices are computed based on the *WCS.rew* file.

Results

We measure the throughput and utilization of the MainServer, throughput of the FlowActuator and the Database. The results are summarized in the following image.

```
Stationary value of the performance measures for WCS_System:

- Value of measure "Mainserver_Throughput":
        2.4235

- Value of measure "Mainserver_Utilization":
        0.0100005

- Value of measure "FlowActuator_throughput":
        1.1969

- Value of measure "DB_Throughput":
        2.44366
```

Figure 13. AEmilia Simulation Results