Modelling and Simulation of a Pump Control System

Miriam Zia
Sadaf Mustafiz

SOCS, McGill University
Fall 2004

Project Description

- Step 1: Create a model for a real-time system BEFORE fault tolerance techniques are applied.
- Step 2: Create a model for the same system, AFTER fault tolerance techniques are applied.
- Step 3: Simulate both models, and observe the dependability of the system in each case.
  - Fault injection will be used to alter the system state.

Fault Tolerant Systems

- Delivery of services despite the presence of hardware or software faults
- Address non-functional requirements like dependability
- Fault tolerance: means of achieving dependability

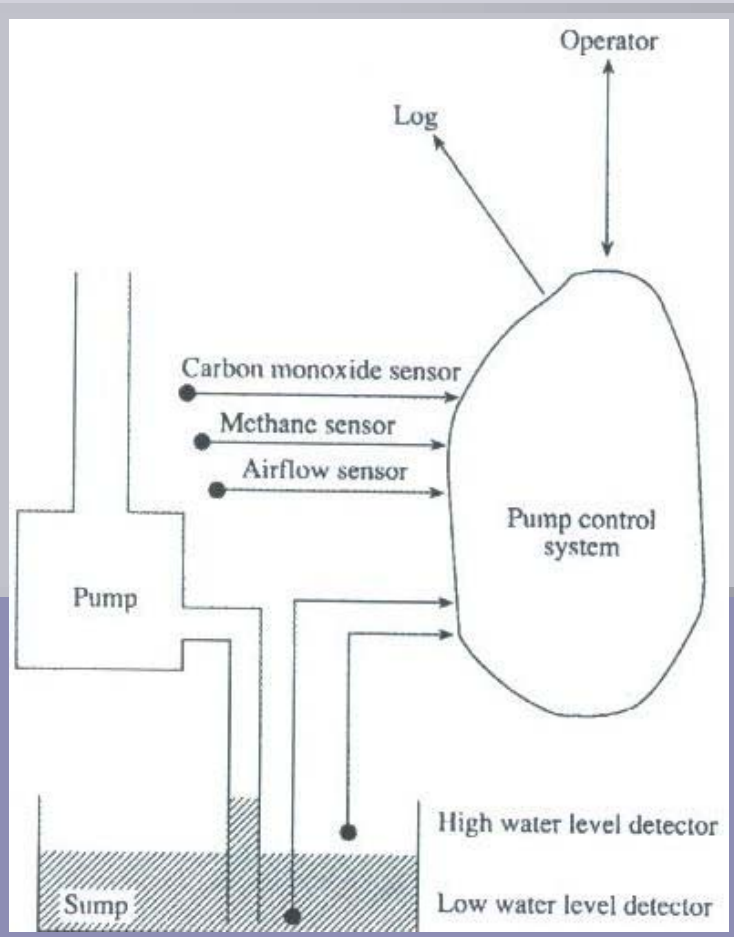
Overview of Presentation

- Pump Control System (problem/ requirements)
- System Architecture to be modeled
- Modelling formalism
- Model of non-fault tolerant system
- Faults in the system, and techniques applied
- Model of fault tolerant system
- Simulation and expected results

Pump Control System: Case Study

- TARDIS:
  - A framework for building timely and reliable distributed systems;
  - Addresses non-functional requirements early in the design process;
- The authors have used this design discipline to make architectural choices to meet the non-functional requirements.

PCS Problem Statement



- water-level >= high-water AND methane-level < critical AND pump-off
  - ➔ water-switch-on
- water-level < = low-water AND pump-on
  - ➔ water-switch-off
- methane-level > critical AND pump-on
  - ➔ methane-switch-off

PCS Non-Functional Requirements

- Metrics that need to be ensured: dependability, timing, security.
- Dependability requirements that need to be satisfied:
  - Reliability
  - Safety

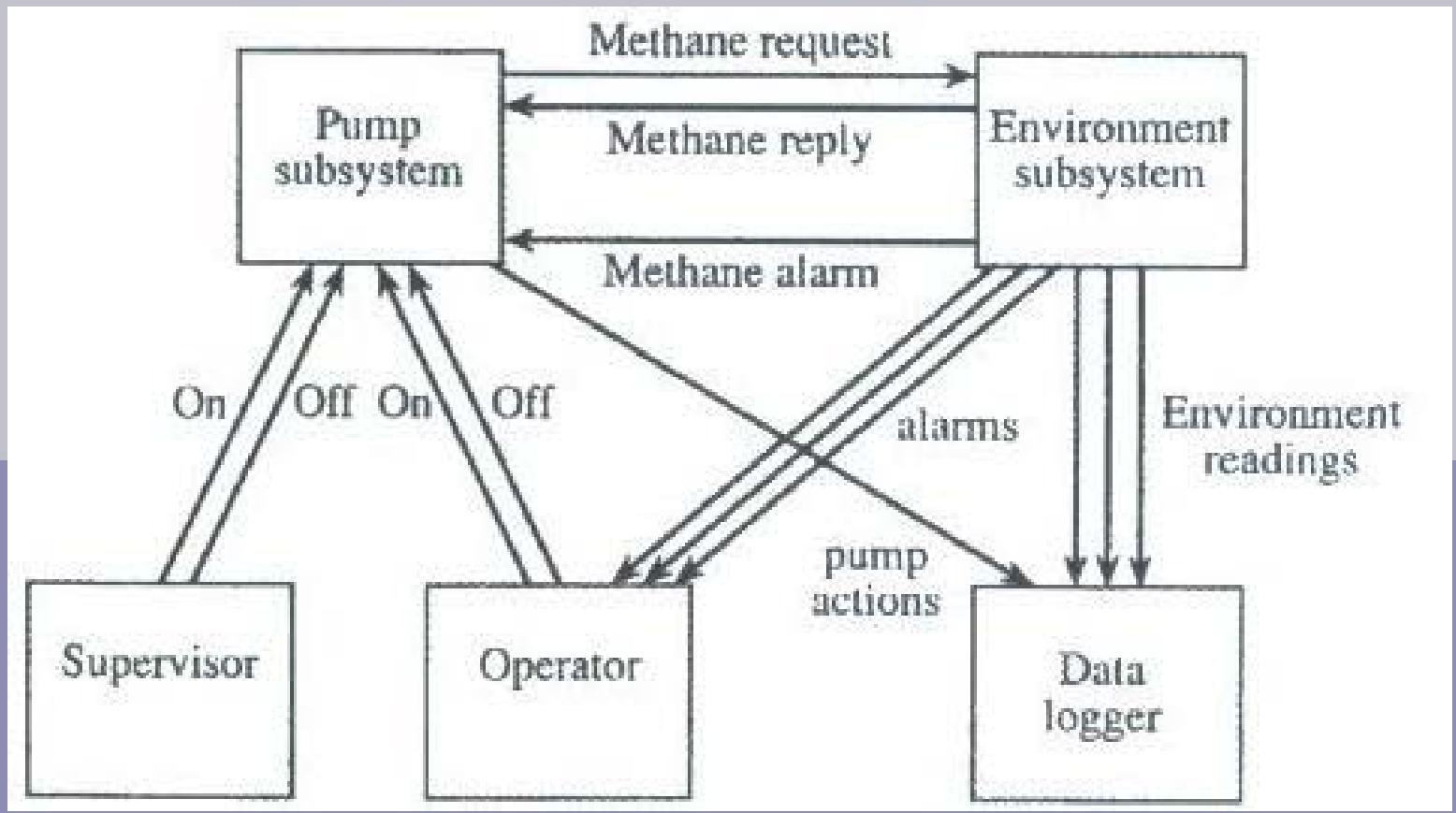
… Reliability

- Measured by:
  - Extent to which the pump operates when it is supposed to.
- Expressed in terms of:
  - The number of work-shifts that can be lost.
  - The paper suggests: 1 in 1000
- [Solution?
  - Running the pump all the time (not safe).]

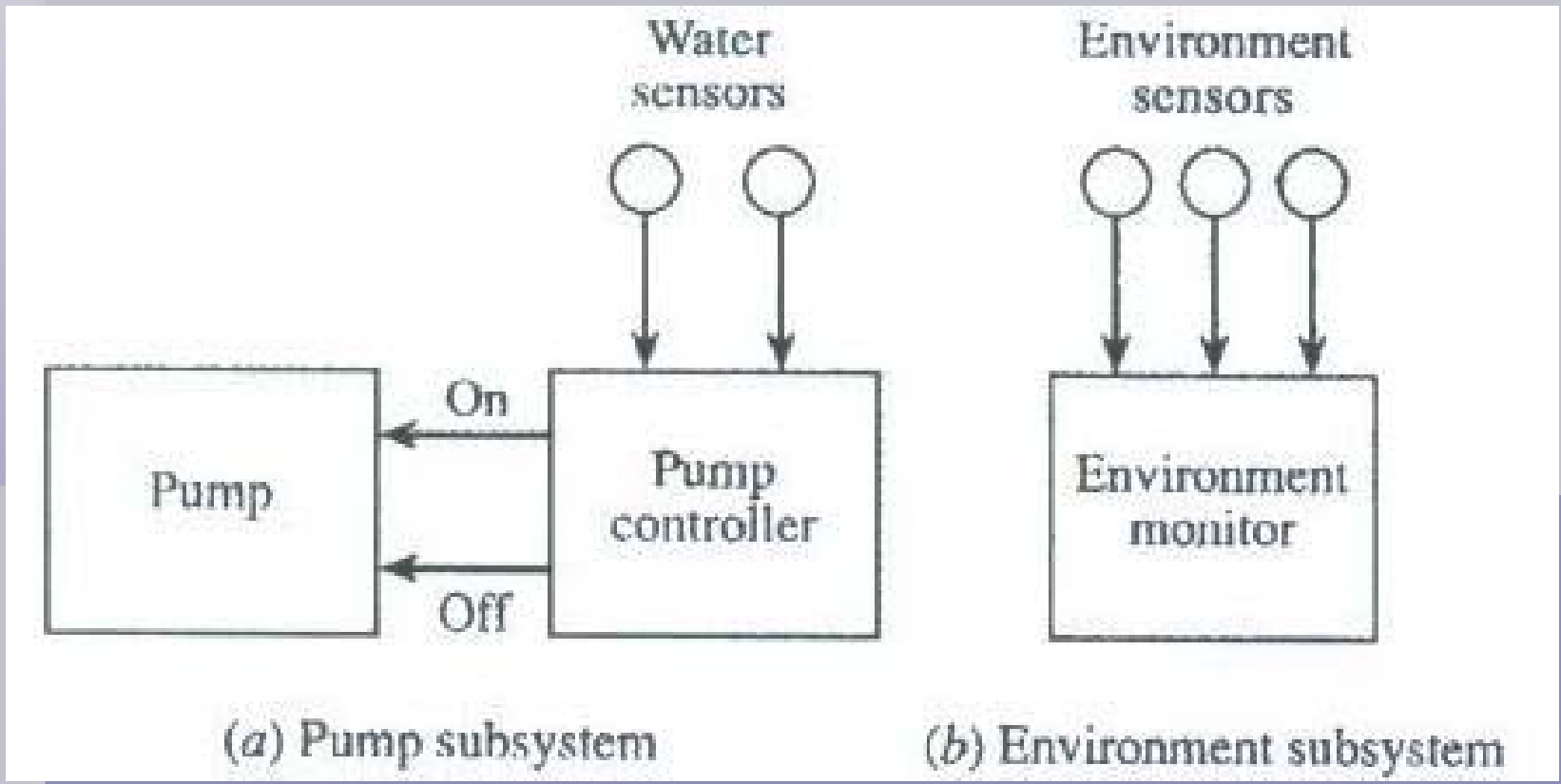
… Safety

- Measured by:
  - The probability of causing an explosion by operating a pump when the methane level is critically high.
- Expressed in terms of:
  - An acceptably low level for that probability.
  - The paper suggests: 10^{-7} (over the system lifetime).
- [Solution?
  - Not running the pump at all (not reliable).]
- The system must meet both requirements by operating between the 2 extremes.

PCS Logical Architecture



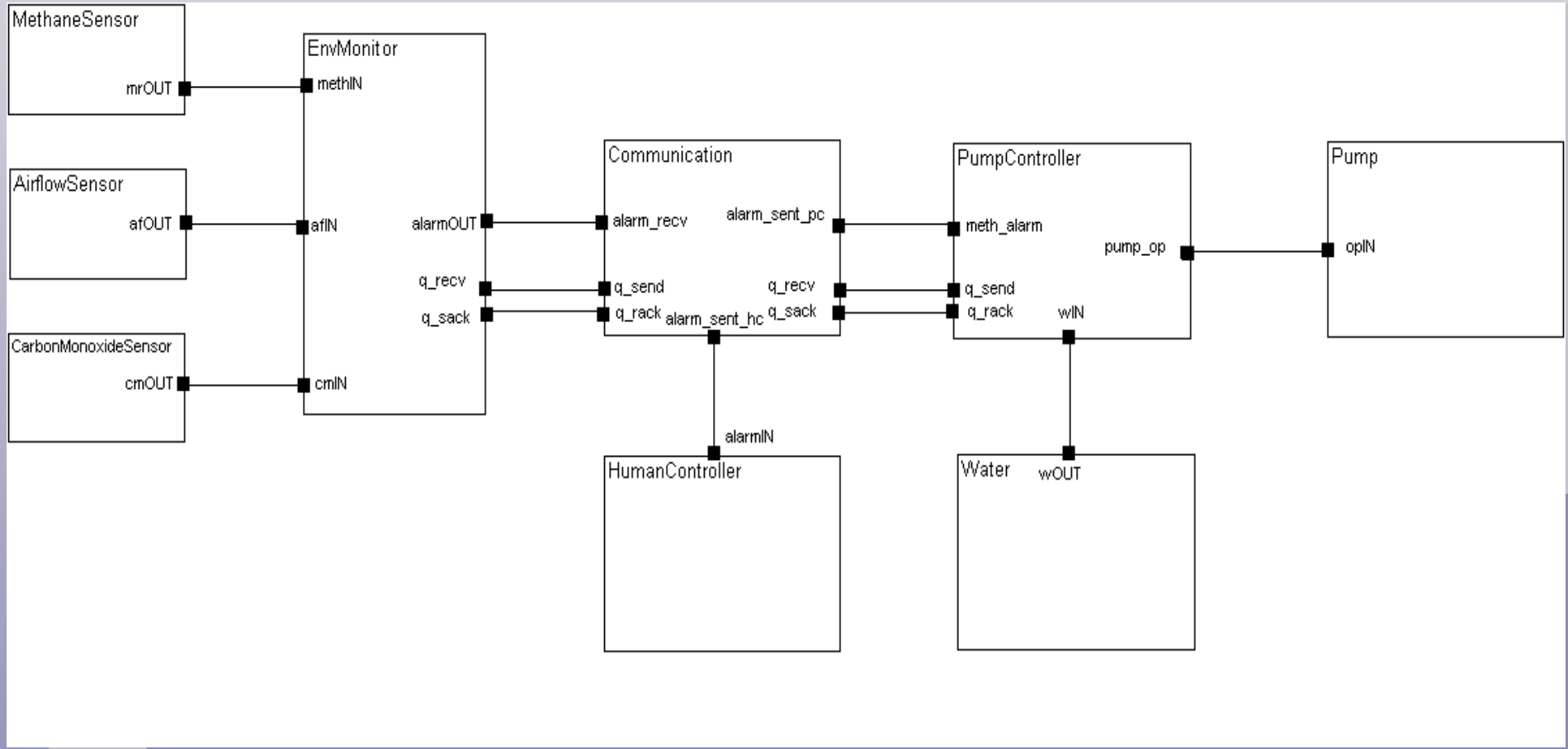
Logical Architecture Refinements



Modelling Formalism: DEVS

- Discrete EVent System specification
- State-based
- Highly modularized
- PCS
  - State change based on external events only
  - No change in system state in between events
  - Time advance (not very nice to model with state charts)

PCS Model


MethaneSensor
mrOUT
EnvMonitor
methIN
AirflowSensor
afOUT
afIN
alarmOUT
q_recv
q_sack
CarbonMonoxideSensor
cmOUT
cmIN
Communication
alarm_recv
alarm_sent_pc
q_send
q_recv
q_rack
alarm_sent_hc
q_sack
alarmIN
HumanController
PumpController
meth_alarm
pump_op
q_send
q_rack
wIN
Water
wOUT
Pump
opIN

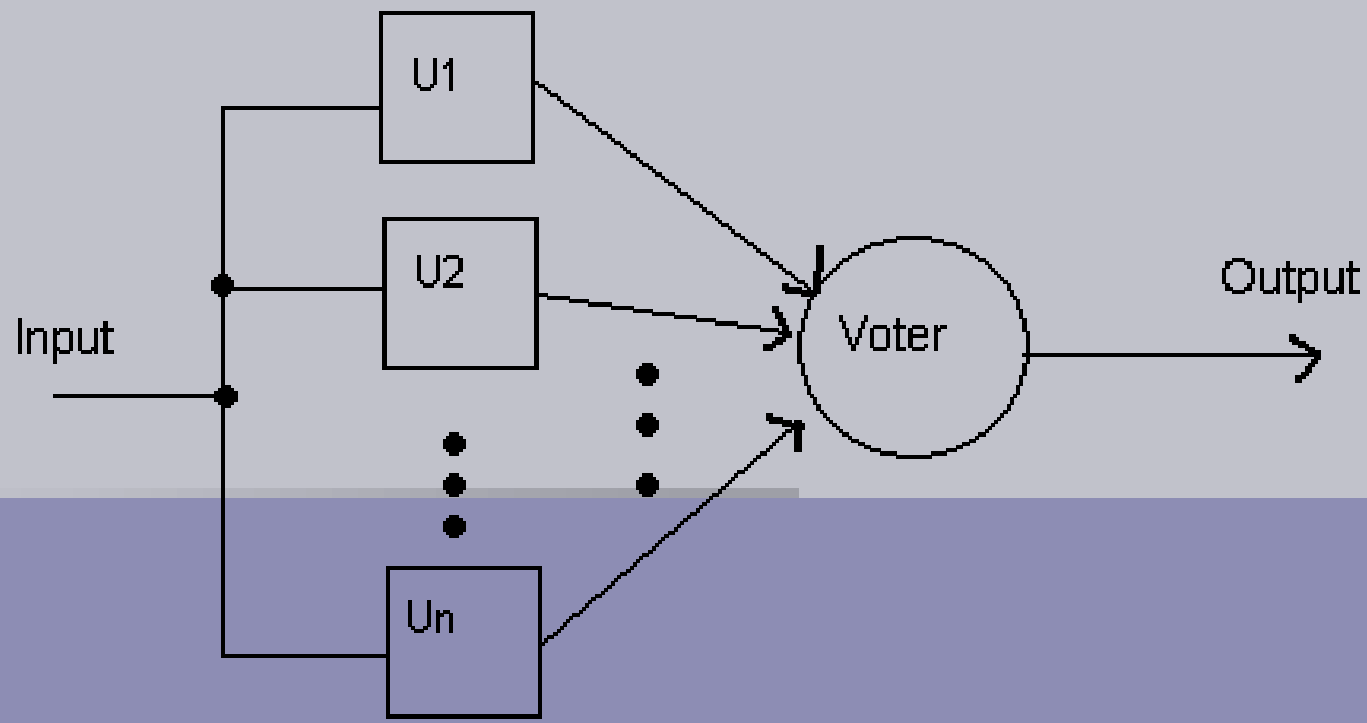
Possible Failure Scenarios

1) *EnvSensors*
   - incorrect values for methane level when asked by the pump subsystem
2) *EnvMonitor*
   - fails to generate an alarm signal when the methane level reaches the danger threshold
3) *Communication*
   - fails to convey the alarm signal to the *PumpController*
4) *PumpController*
   - fails to switch off the *Pump* when it receives the alarm

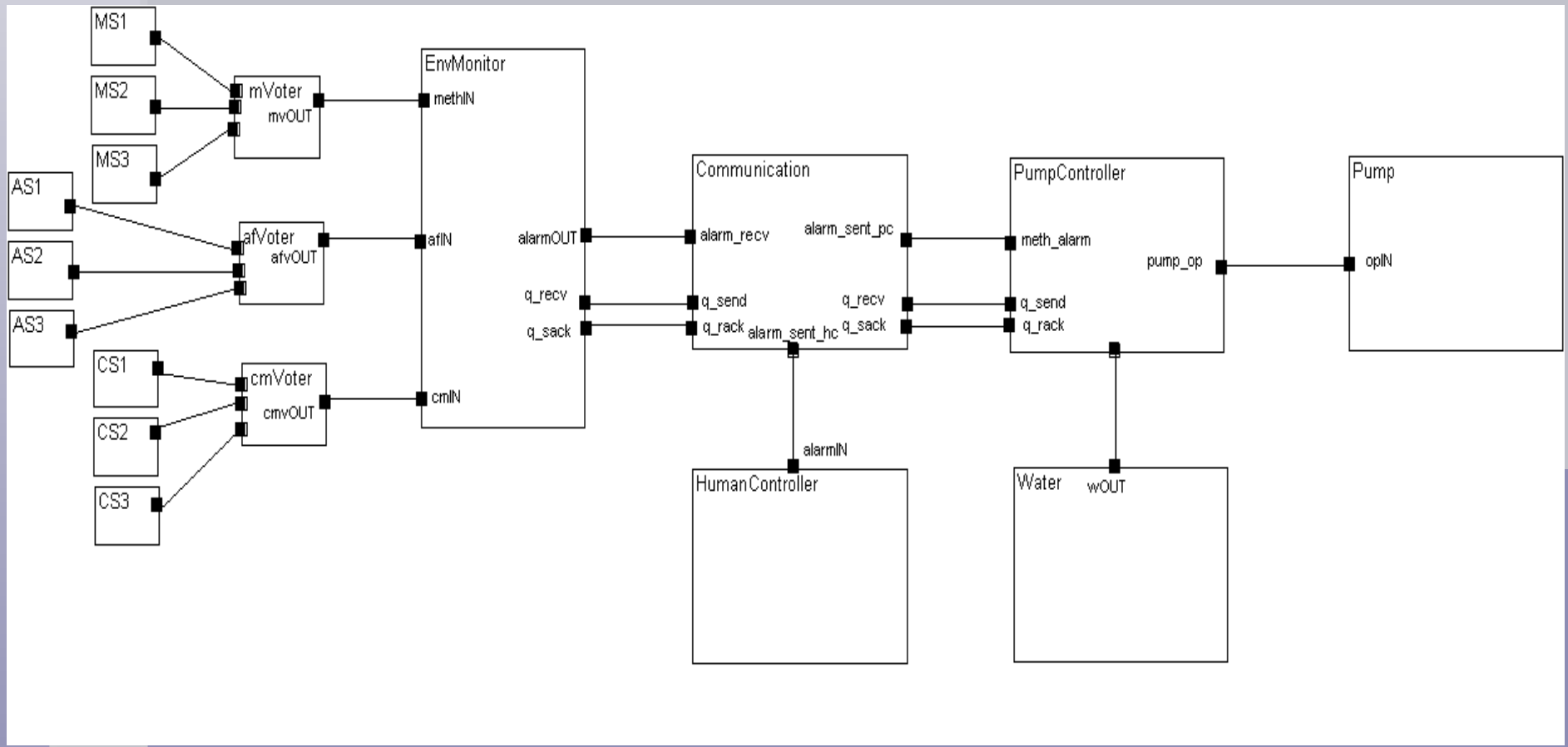
Failure Scenario: EnvSensors

- Sensors don't fail silent: give erroneous readings rather than none at all.
- Only means of failure detection is through replication.
  - ***Solution:*** 3 sets of sensors and use N-modular redundancy (NMR) for failure detection.

How does NMR work again?



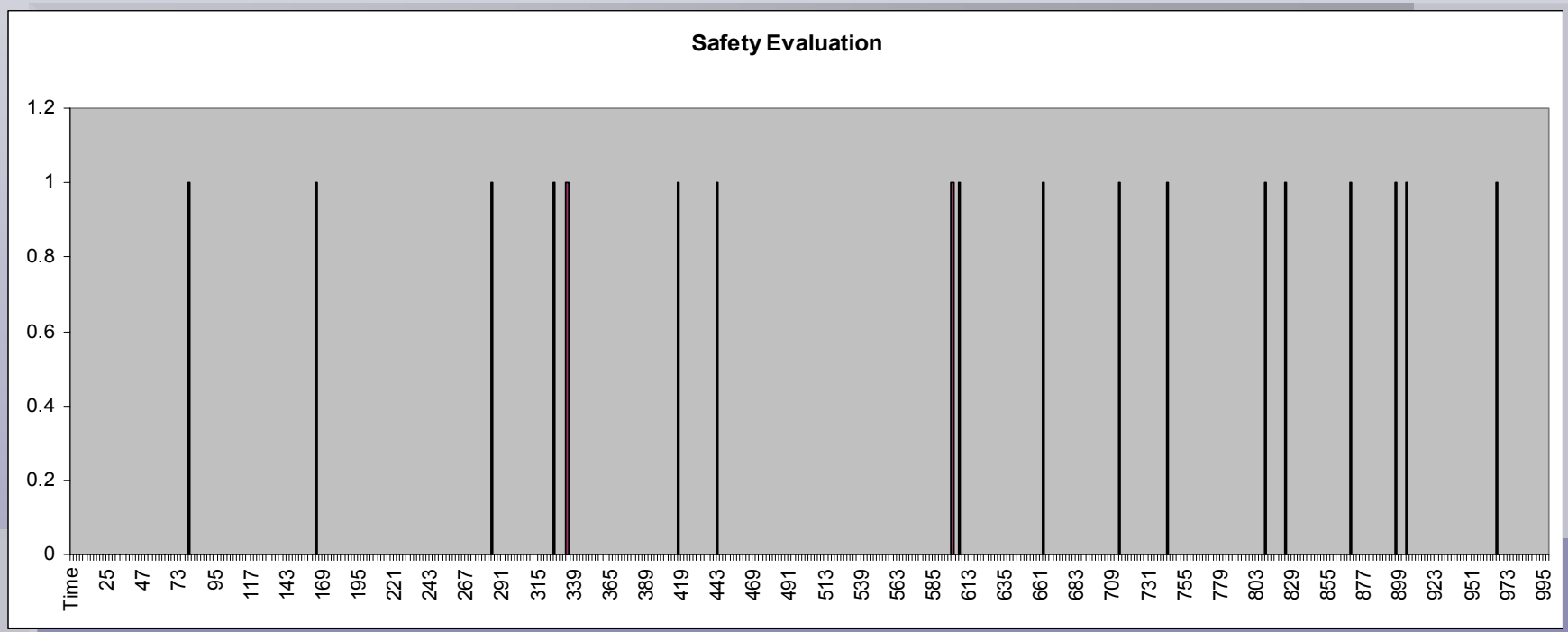
Fault Tolerant PCS Model



Simulation

- Python DEVS Simulator
- Ran the simulation for 1000 seconds
- We kept track of safety performance whenever a methane reading was generated by the methane sensor.
- Similarly, the performance levels in terms of reliability can be simulated and analyzed.

Simulation Results



- 454 methane readings over 1000 seconds
- Safety requirement failed: 18 times
- Probability of failure: 3.97%

To Be Completed…

- Gather results for the reliability in original model.
- Implement the DEVS model for the fault-tolerant system
- Gather simulation results
- Compare performance of the modified model to the original model
  - Check for improvements in dependability
- Complete the report

Extensions

- Extend model with NMR techniques for environment monitor and pump controller systems.
- External fault injector
  - FI DEVS to send external events to components
- Human controller: active DEVS
- Extend model to simulate other performance metrics like timeliness, security, etc.

… so what's the point?

- Model system behaviour to observe where faults may occur
- Test whether fault tolerance techniques improve system performance
  - Dependability, timeliness
- Optimize system
- Reduce cost of development

References

- A.Burns, M.Lister; “A framework for building dependable systems”; The Computer Science Journal, Vol.34, No.2, 1991.
- COMP-522: Modelling and Simulation
  - www.cs.mcgill.ca/~cs522
- Hans Vangheluwe; “The Discrete EVent System specication (DEVS) formalism”.
- Project report: www.cs.mcgill.ca/~smusta4/522Project