The Weather System

A Low-Cost, Energy Efficient Weather Broadcasting System

Kelvin Lin
Weilin Hu
Karl Knopf
Jie Luo

October 22, 2016

Contents

Module Overview

The following document shows the decomposition hierarchy of Team 1's weather system.

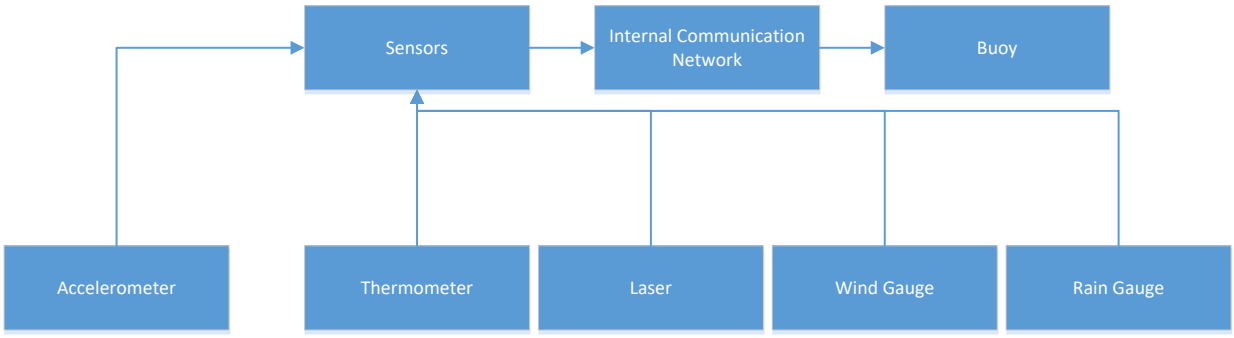

Figure 1: System Interactions

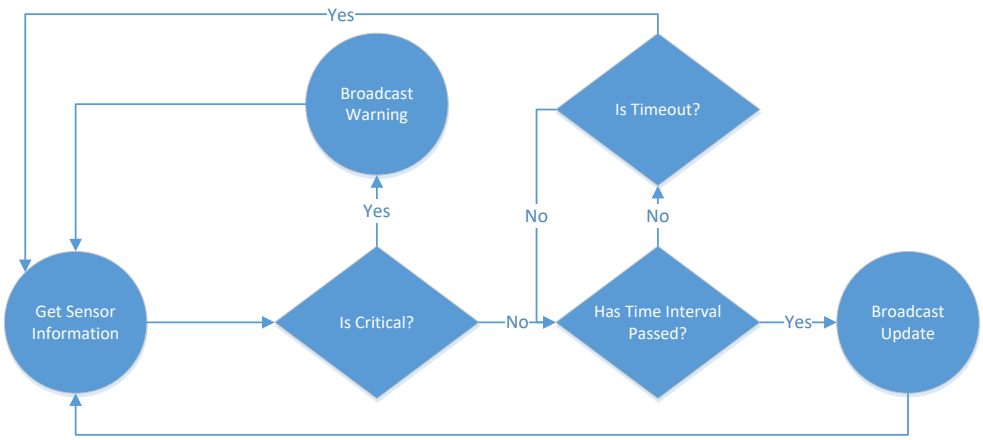

Figure 2: Dataflow

As shown from the diagrams above, the weather system consists of three main components: the sensors, the internal communication network, and the buoy. The sensors (i.e. the rain gauge, wind meter, laser, and thermometer) are located on the buoy. They all collect data about the respective surrounding, and send the data to the internal communication network. The internal communication network contains a array that stores each sensor's most current read data. The buoy sequentially checks each element of the array, and the buoy checks the criticality of the weather based on pre-programmed "danger" levels in the software. Note that this is done continuously. If there are no critical levels then after a pre-

determined time interval has passed, the buoy will find an available radio channel and broadcast the weather information. Otherwise, if there is a critical value, then the buoy will immediately send a weather update, and then periodically send updates at a shorten time interval until the critical condition clears.

Note: In figure 1, an arrowing pointing from A → B denotes that A sends messages to B.

Requirements

Functional Requirements (Condensed Due to Space Limitations)

Requirement #: 1 **Type: Functional Requirement**

The Buoy shall receive data from the Internal Communications Network.

Fit Criterion: The Buoy is able to get data from the Internal Communications Network after sending a request without errors.

Rationale: The Internal Communications Network mediates all internal communication between the systems. If the Buoy cannot communicate with the network, then it cannot help save lives.

Requirement #: 2 **Type: Functional Requirement**

All sensors shall be able to send information to the Internal Communications Network.

Fit Criterion: The sensors are able to send information to the Internal Communications Network without errors.

Rationale: The Internal Communications Network mediates all internal communication between the systems. If the sensors cannot communicate with the network, then the Buoy cannot help save lives.

Requirement #: 3 **Type: Functional Requirement**

The Buoy shall be able to find empty channels to broadcast messages.

Fit Criterion: The buoy can find the least full channel to broadcast messages.

Rationale: Broadcasting to a full channel will result in a meaningless message.

Requirement #: 4 **Type: Functional Requirement**

The Buoy shall be able to scan received data for indications of dangerous environments.

Fit Criterion: The buoy can determine the danger level of its environment based on its data.

Rationale: In order to save lives, the buoy must be able to detect if the environment is dangerous.

Non-Functional Requirements (Condensed Due to Space Limitations)

Requirement #: 5	**Type: Non-Functional Requirement**

The system shall be available to users using a variety of devices.

Fit Criterion: The system is available to users who are using Windows, Mac, and Linux.

Rationale: Different people sailing may use different systems,

Requirement #: 6	**Type: Non-Functional Requirement**

The system shall be precise to within a specified precision.

Fit Criterion: The output from the testing condition, and the specified testing condition should deviate by no more than the specified precision.

Rationale: If the system is not precise, then the data is not reliable.

Requirement #: 7	**Type: Non-Functional Requirement**

The system shall not have an adverse impact on the environment.

Fit Criterion: The system does not decrease the livelihood of local animals, and it does not change the chemical composition of the water.

Rationale: Damage to the environment can incur legal costs.

Requirement #: 8	**Type: Non-Functional Requirement**

The response time of the system shall be no more than a specified value.

Fit Criterion: The system provides output after a specified time interval.

Rationale: Outdated information is not useful for users.

Module Specification

CLASS: Buoy

The Buoy class represents the actions taken by the control module of the buoy. It can receive data from sensors, and transmit data to users.

INTERFACE

USES

InternalCommunicationsNetwork, Radio

ACCESS PROGRAMS

Buoy()

Initializes any field variables

receive(i: Integer): void

Gets data with id i from the InternalCommunicationsNetwork.

send(): String

Returns a message with the processed data from the InternalCommunicationsNetwork.

isCritical(): Integer

Checks the data received and returns the danger level.

CLASS: InternalCommunicationsNetwork

The InternalCommunicationsNetwork represents the internal network used by the Buoy to communicate with the sensors.

INTERFACE

USES

None

ACCESS PROGRAMS

InternalCommunicationsNetwork()

Initializes any field variables

put(i: Integer, data: Object): void

Puts an object data into the internal communication network with id i.

get(i: Integer): Object

Returns the object data with id i.

getAll(): Object[]

Returns all of the data in the internal communication network.

CLASS: Laser

The laser class simulates the behaviours of a laser system to determine the current visibility conditions. In a physical system, it will obtain and process the information obtains from a laser system, measuring the amount of refraction in the air.

INTERFACE

USES

InternalCommunicationsNetwork

ACCESS PROGRAMS

Laser()

Initializes any field variables

send():double

Sends a real number, representing the detected refraction, to the Internal Communication Channel.

receive(): void

Receives input from the physical laser.

CLASS: Thermometer

The thermometer class simulates the behaviours of a real thermometer. In a physical system, it will obtain and process the information obtains from a thermometer.

INTERFACE

USES

InternalCommunicationsNetwork

ACCESS PROGRAMS

Thermometer()

Initializes any field variables

send():double

Sends a real number, representing the detected temperature, to the Internal Communication Channel.

receive(): void

Receives input from the physical thermometer.

CLASS: Radio

The radio class simulates the channels available on the FM radio scan.

INTERFACE

USES

None

ACCESS PROGRAMS

Radio()

Initializes any field variables.

scan(i: Integer): Boolean

Indicates whether channel i is open.

lockOn(i: Integer): void

Indicates that channel i is in use.

release(i: Integer): void

Indicates that channel i is no longer in use.

size(): Integer

Returns the total number of channels in the frequency.

CLASS: Rain Gauge

The thermometer class simulates the behaviours of a real thermometer. In a physical system, it will obtain and process the information obtains from a thermometer.

INTERFACE

USES

InternalCommunicationsNetwork

ACCESS PROGRAMS

RainGuage()

Initializes any field variables

send():double

Sends a real number, representing the detected level of precipitation, to the Internal Communication Channel.

receive(): void

Receives input from the physical thermometer.

CLASS: Wind Gauge

The wind gauge class simulates the behaviours of a real wind gauge. In a physical system, it will obtain and process the information obtains from a wind gauge.

INTERFACE

USES

InternalCommunicationsNetwork

ACCESS PROGRAMS

WindGauge()

Initializes any field variables

send():Wind

Sends a real number, representing the detected wind speed and an integer representing the calculated wind direction, to the Internal Communication Channel.

receive(): void

Receives input from the physical wind gauge.

Assumptions

1. The temperature is between -30 degrees Celsius to 60 degrees Celsius.
2. The possible precipitation range is between 0 mm and 400 mm.
3. The wind speed ranges between 0 km/hr to 200 km/hr.
4. The wind can only come from North, South, East, West, North-East, North-West, South-East, South-West, or no wind.
5. The laser returns a value between 0 and 100, which represents how much light reflected back.
6. There are 5 danger levels (from most severe to least severe): hurricane, storm, gale, little danger, no danger.
7. The travellers understand the metric system.
8. The travellers have access to a computer system with Java.

Appendix

Error Codes

1. IOException: The value is not the correct expected value.
2. IndexOutOfBoundsException: The index specified for the array is not appropriate.

Libraries Used

1. Standard Java Libraries (i.e. Random)
2. JSwing Java Library
3. Junit

References

Not applicable for this documentation.