**COMP 2920 : Software Architecture & Design**

**Class Activity #4**

Due date: 11th Nov 2017 Weight: 1%

Q1. Research to look for a recent large software system that is behind the schedule, over budget, or failed to perform as expected. What factors were blamed? How could this failure have been avoided?

Q2. Prepare a list of classes that you would expect the following system to handle:

1. A program to compute and store bowling scores

Player

Session - Round

Score

Turn

1. A controller for a video cassette recorder

Function

Play

Record

Write

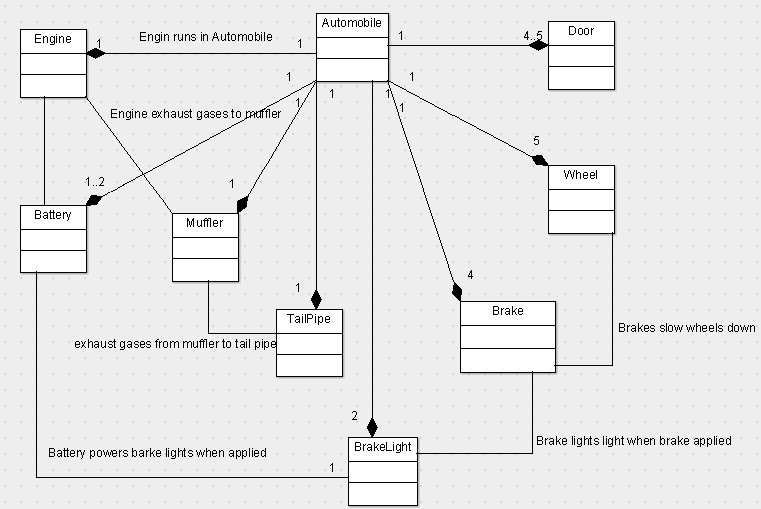
Stop

Burn  
Settings

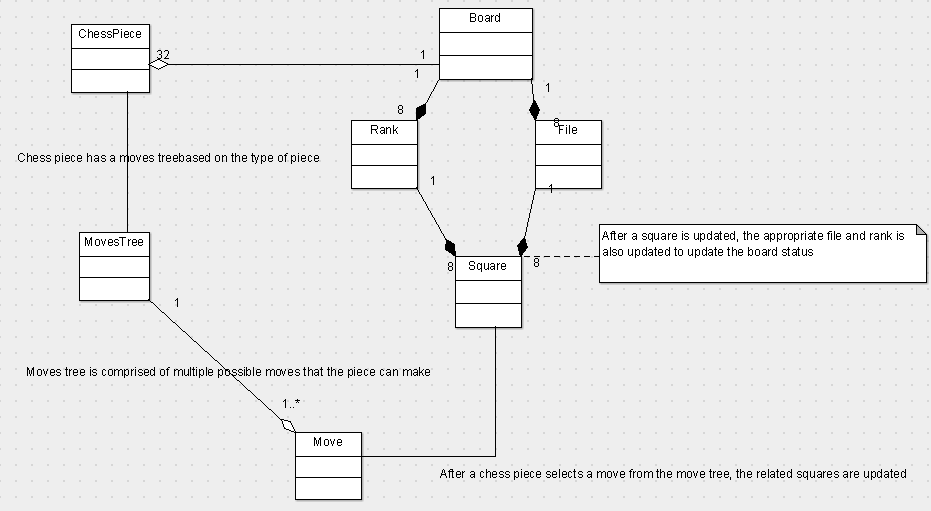
Receiver

Q3. Prepare a class diagram using the following group of classes. You need to define associations and generalizations to the diagram.

Automobile, engine, wheel, brake, brake light, door, battery, muffler, tail pipe



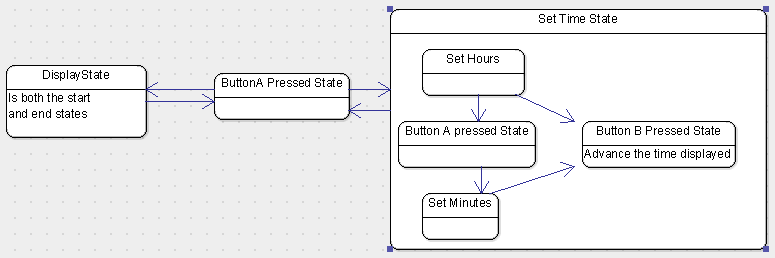
Chess piece, rank, file, square, board, move, tree of moves



Q4. Explain the ad hoc approach of software development?

In ad hoc software development, not a lot of emphasis is given on documentation and setup of the development entities. It focuses more on doing as fast as possible and is best suitable for small scale software projects.

Q5. A simple digital watch has a display and two buttons to set it, the A button and the B button. The watch has two modes of operation, display time and set time. In the display time mode, hours and minutes are displayed, separated by a flashing colon. The set time mode has two modes, set hours and set minutes. The A button is used to select modes. Each time A is pressed, the mode advances in sequence: display, set hours, set minutes, display etc. Within the sub modes, the B button is used to advance the hours or minutes once each time it is pressed. Buttons must be released before they can generate another event. Prepare a state diagram of the watch.



Q6. Read the description below and do an architectural design.

Software for sea buoys support for navigation at sea. There exists a collection of free floating buoys that provide navigation and weather data to air and ship traffic at sea. The buoys collect air and water temperature, wind speed, and location data through a variety of sensors. Each buoy may have a different number of wind and temperature sensors and may be modified to support other types of sensors in the future. Each buoy is also equipped with a radio transmitter (to broadcast weather and location information as well as an SOS message) and a radio receiver (to receive requests from passing vessels). Some buoys are equipped with a red light, which may be activated by a passing vessel during sea-search operations. If a sailor is able to reach the buoy, he or

she may flip a switch on the side of the buoy to initiate an SOS broadcast.

Software for each buoy must:

* maintain current wind, temperature, and location information; wind speed readings are taken every 30 seconds, temperature readings every 10 seconds and location every 10 seconds; wind and temperature values are kept as a running average;
* broadcast current wind, temperature, and location information every 60 seconds;
* broadcast wind, temperature, and location information from the past 24 hours in response to requests from passing vessels;
* activate or deactivate the red light based upon a request from a passing vessel;
* Continuously broadcast, an SOS signal after a sailor engages the emergency switch.

Answer the following architecture design questions. Motivate for your choices and state

Your assumptions.

1. Name two important quality attributes that the architecture design should consider. Justify your answer.

* Modifiable since it has a variety of sensors that can be exchanged and replaced
* Continuous broadcast of SOS signals until heard to increase the chances of someone receiving the signal and responding to it
* High range in terms of distance to broadcast and receive signals to transmit signals to longer distances and be able to read weak signals
* Continuously running and record results after certain periods for a long period to maintain the paths history and running average of wind and temperature

1. What architectural style and control pattern will you use in your design?

I would think a combination of Component based architecture and peer-to-peer would be suitable for this. The component-based system would allow for the modifiability and changeability of the sensors whereas the Peer to peer would allow connection and transmission between buoys and sailing ships. The buoys would be able to connect and transmit data over a longer distance rather than one buoy transmitting alone.

The control would be an endless loop that continuously reads and records sensor data after a defined period unless interrupted by an SOS event or a ship search event in which case it would give priority to broadcasting an SOS message and controlling the search lights.

1. Specify the type [1] (event-driven, periodic, or demand-driven) for each I/O interface task in your design.

For recording wind and temperature data, a periodic system can be used where the System clock would run endlessly recording data and calculating the averages. But some events such as when passer by ships connect to it and request data, the periodic loop would be interrupted and priority should be given to the ships requests. Also, when an SOS broadcast is to be made, this would be given highest priority for a long period of time.

The UI to interact with the buoy would mainly include a BIG RED button to publish an SOS message and make it easier for the person who wants it to be broadcast. Otherwise, to connect to the buoy, a wireless connection would be enough to access the storage and data from it. Therefore, radio waves at a specific frequency would be broadcast to connect and retrieve data from the buoy. The output would also be in form of radio waves broadcast at a specific frequency.

1. Structure the system into subsystems. For each subsystem, structure the system into concurrent tasks (this is a real-time system!). Make decisions about the characteristics of messages, in particular, whether they are asynchronous or synchronous (with or without reply). Show the dynamic view of your architecture design.

The buoy system would be broken down into three major sub-systems: The data Collection and storage, SOS broadcast and Data broadcast.

The data collection and storage would be synchronized to run endlessly and synchronize with the Data receive and broadcast system. Each sensor would be running concurrently to make sure that the timing is consistent and uniform. The data broadcast could be demand driven i.e. only occurs when a connected passer by ship requests data from the buoy and the data collection system would continue running in the background when the data broadcast and receive is called.

The SOS broadcast would be different in that when triggered, it would take priority over the rest and the other systems would be stopped to allow resources for the SOS to be sent.