Albert Wallace
aew0024
April 8, 2013
COMP3700 Homework 5

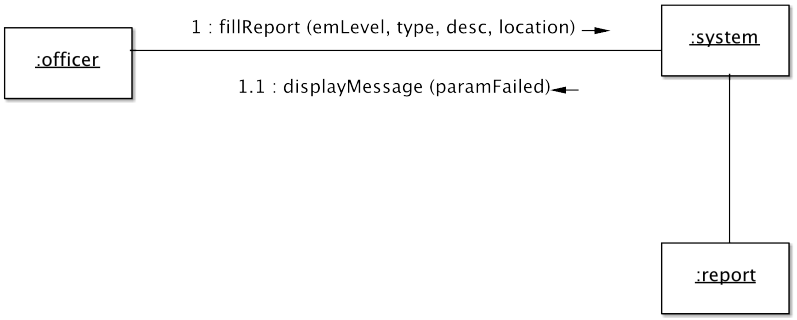
[[[A]]]

a) [An electronic chess companion. The system consists of a chess board with a built-in computer, lights, and membrane switches. The human player registers moves by pressing chess pieces on the board, activating membrane switches mounted under each square. The computer indicates moves through lights also mounted under each square. The human moves the chess pieces for the computer. The computer should make only legal moves, should reject attempted illegal human moves, and should try to win.]
    a) event-based (maybe interrupt-driven) control: in this case, there are multiple external inputs which can generate multiple external events, and the central control must respond to the external event through the other subsystems. In this case, the response would be through the lights to display moves, and the moves themselves created in response to a human. As such, the human's moves are the external stimuli, and the membrane switches activated are the external controls. The external controls trigger an interrupt, which tells the system to perform the rest of its duties (i.e. to make a move in response to the human, if possible).
b) [An airplane flight simulator for a video game system. The video game system has already been implemented and consists of a computer with joystick and pushbutton inputs and an output interface for a color television. Your job is to develop the software for the computer to display the view from the cockpit of an airplane. The joystick and pushbutton control the airplane. The display should be based on a terrain description stored in memory. When your program is complete, it will be sold on cartridges that plug into the video game system.]
    a) event-based (maybe interrupt-driven) control: in this case, interrupts are required, as the game absolutely must run in real-time. The joystick and pushbutton inputs are the external controls, and when pressed they generate interrupts which the game code responds to & changes the simulation accordingly.
c) [A floppy disk controller chip. The chip is going to use a microprogram for internal control. You are concerned with the microprogram. The chip bridges the gap between a computer and a floppy disk drive. Your portion of the control will be responsible for positioning the read/write head and reading the data. Information on the diskette is organized into tracks and sectors. Tracks are equally spaced circles of data on the diskette. Data within a track is organized into sectors. Your architecture will need to support the following operations: Find track 0, find a given track, read a track, read a sector, write a track, and write a sector.]
    a) The controller chip is likely a model of centralized control. Once the commands are passed from the other parts of the system, the controller will manage all other subsystems of the FDD. It tells the servos what to do, and it interprets the raw data provided from the read/write heads.

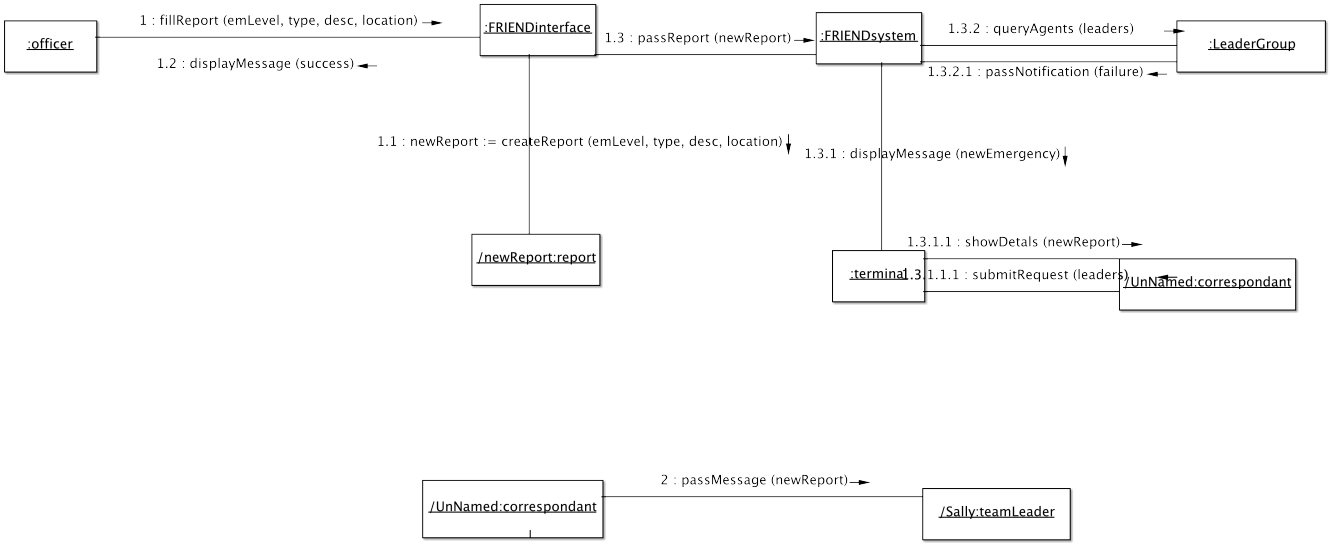
d) [A sonar system. You are concerned with the portion of the system that detects undersea objects and computers how far away they are (range). This is done by transmitting an acoustic pulse and analyzing any resulting echo. A technique called correlation is used to perform the analysis, in which a time-delayed copy of the transmitted pulse is multiplied by the returned echo and integrated for many values of time delay. If the result is large for a particular value of time delay, it is an indication that there is an object with a range that corresponds to that delay.]

a) Call-return models. In a way, the acoustic pulse is the call, and the return is the echo, which influences the next step in the process of determining the distance based on the return itself.

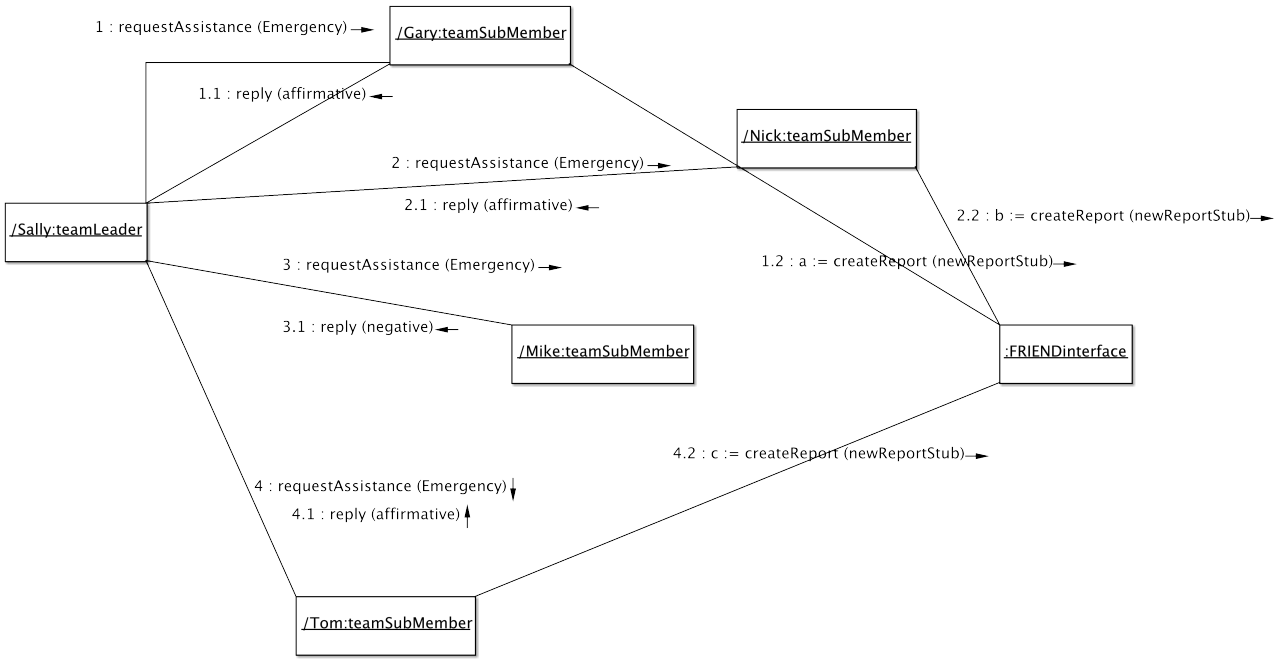
[[[B]]]
Scenario 1:

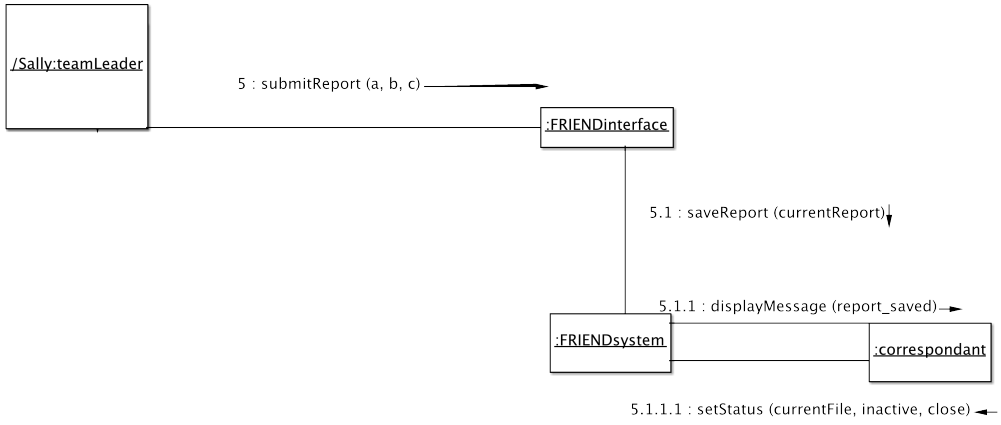


Scenario 2:


:officer
1 : fillReport (emLevel, type, desc, location)
1.2 : displayMessage (success)
:FRIENDinterface
1.3 : passReport (newReport)
:FRIENDsystem
1.3.2 : queryAgents (leaders)
1.3.2.1 : passNotification (failure)
:LeaderGroup
1.1 : newReport := createReport (emLevel, type, desc, location)
1.3.1 : displayMessage (newEmergency)
/newReport:report
1.3.1.1 : showDetals (newReport)
:terminal
1.3.1.1.1 : submitRequest (leaders)
/UnNamed:correspondant
2 : passMessage (newReport)
/UnNamed:correspondant
/Sally:teamLeader

Scenario 3:


1 : requestAssistance (Emergency)
/Gary:teamSubMember
1.1 : reply (affirmative)
/Nick:teamSubMember
2 : requestAssistance (Emergency)
2.1 : reply (affirmative)
2.2 : b := createReport (newReportStub)
/Sally:teamLeader
3 : requestAssistance (Emergency)
1.2 : a := createReport (newReportStub)
3.1 : reply (negative)
/Mike:teamSubMember
:FRIENDinterface
4.2 : c := createReport (newReportStub)
4 : requestAssistance (Emergency)
4.1 : reply (affirmative)
/Tom:teamSubMember


/Sally:teamLeader
5 : submitReport (a, b, c)
:FRIENDinterface
5.1 : saveReport (currentReport)
5.1.1 : displayMessage (report_saved)
:FRIENDsystem
:correspondant
5.1.1.1 : setStatus (currentFile, inactive, close)