University of Waterloo

Final Examination

E&CE 355
Software Engineering

10/12/2004

Instructor

Kostas Kontogiannis

This exam has seven questions, and is marked out of 110 points. The examination is 180 minutes long. This is a closed book exam. No additional material or calculators are allowed. Good luck.

Elevator Problem

A product is to be installed to control elevators in a building with m floors. The problem concerns the logic required to move elevators between floors according to the following constraints:

- Each elevator has a set of m buttons, one for each floor. These illuminate when pressed and cause the elevator to visit the corresponding floor. The illumination is canceled when the elevator visits the corresponding floor.

- Each floor, except the first floor and top floor has a panel of two buttons, one to request and up-elevator and one to request a down-elevator. These buttons illuminate when pressed. The illumination is canceled when an elevator visits the floor and then moves in the desired direction.

- When an elevator has no requests, it remains at its current floor with its doors closed.

Q1. Pertaining to the elevator problem use UML to specify the following artifacts:

1. Specify the Use Case for the following scenario

    - Passenger presses floor button
    - Elevator system detects floor button pressed
    - Elevator moves to the floor
    - Elevator doors open
    - Passenger gets in and presses elevator button
    - Elevator doors closes after 5 seconds
    - If the Passenger does not press any button after the Passenger enter the elevator the door closes after 15 seconds and the elevator car stays idle on the floor. If the elevator is idle on a floor with a passenger inside the passenger can press the open-door button and the door opens
    - Elevator moves to required floor
    - Elevator doors open
    - Passenger gets out
    - Elevator doors closes after 15 seconds and elevator stays idle on the floor

2. Draw the class diagram for the Elevator System

3. Draw the Message Sequencing Diagrams pertaining a) to an elevator button and b) to the panel of the two floor buttons

4. Draw the Collaboration Object Diagrams pertaining a) to an elevator button and b) to the panel of the two floor buttons

5. Draw the State Diagrams pertaining to a) to the elevator car, b) the elevator doors.

Q2. Design Basics

1. List in the order of significance (from better to worse) the different types of cohesion. What is Procedural Cohesion and how it differs from Temporal Cohesion.

2. List in the order of significance (from better to worse) the different types of coupling. What is Common Coupling and how it differs from Control Coupling.

3. Provide a definition of Software Architecture, and discuss briefly the issues addressed by Architecture Design.

Q3. Architectural Styles

1. Please provide the following
   a) What is an Architectural Style
   b) What is an OO Framework
   c) What are the basic categories of OO Frameworks and how frameworks in these categories can be reused?

2. Consider a system that allows for sets of *n* data items to be read by a sensor every k seconds. The system sorts each data set in ascending order, computes the average value of data in each set and outputs the sets according to the order of the average value in each set. For example all the data in the data set with the highest average value will be output first, all the data in the data set with the second highest average value second, etc. Please provide the architecture of the above system using:
   a) Data Flow architectural style
   b) Implicit Invocation style
   c) Object Oriented style

Q4. Design Patterns

1. Please give the following characteristics of the Strategy pattern:
   a) Intent
   b) Applicability
   c) Basic Structure or use (you can provide either a class diagram or example pseudocode of its use)
   d) Consequences

2. Please give the following characteristics of the Abstract Factory pattern:
   a) Intent
   b) Applicability
   c) Basic Structure or use (you can provide either a class diagram or example pseudocode of its use)
   d) Consequences

3. Imagine that you are implementing an output text stream library. All streams in your library will share the same abstract interface. Examples of concrete streams in the library are TextFileStream, ClipBoardStream and LinePrinterStream. Additionally, the library should allow the user to extend a given stream with additional capabilities such as buffering text, scrambling text and performing textual analysis while writing the text.
   The design should allow the library user to add one or more of these additional capabilities to a concrete stream. There should be no difference for the application code whether it writes to a bare concrete stream or one with any combination of the additional capabilities.

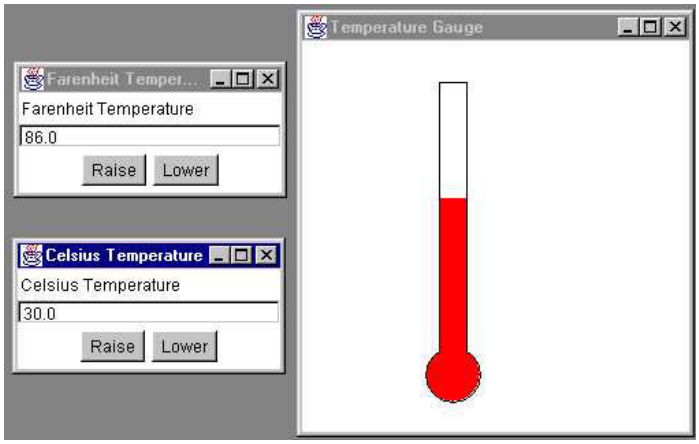
   a) What design pattern could be used to achieve this flexibility?
   b) Please explain the library design by giving a class diagram.

4, Imagine that you are implementing a file system. The main abstractions in your design would be files and directories. Directories can contain zero or more files or directories. You want to treat directories and files in uniform way, e.g., both will have name and will provide operations to *stream* content *in* and *out*, and to *list* children.

   a) What design pattern could be used to achieve this design?
   b) Please explain your design by giving a class diagram.

Q5. Farheneit to Celcius Converter problem

Consider the following application that allows the user to adjust the value of the temperature using the *Raise* or *Lower* buttons in the Celsius or Fahrenheit section of the GUI illustrated below. When a value is inserted the corresponding value in the other scale is displayed. Also the bar in the temperature gauge on the right is also adjusted accordingly.



1) Which design pattern is most appropriate to be used in this application?

2. Please explain the GUI design by giving a class diagram

3. What architectural style is most appropriate for this application? Draw the architecture of such a system using the architecture style you consider most appropriate. Please explain your design decisions.

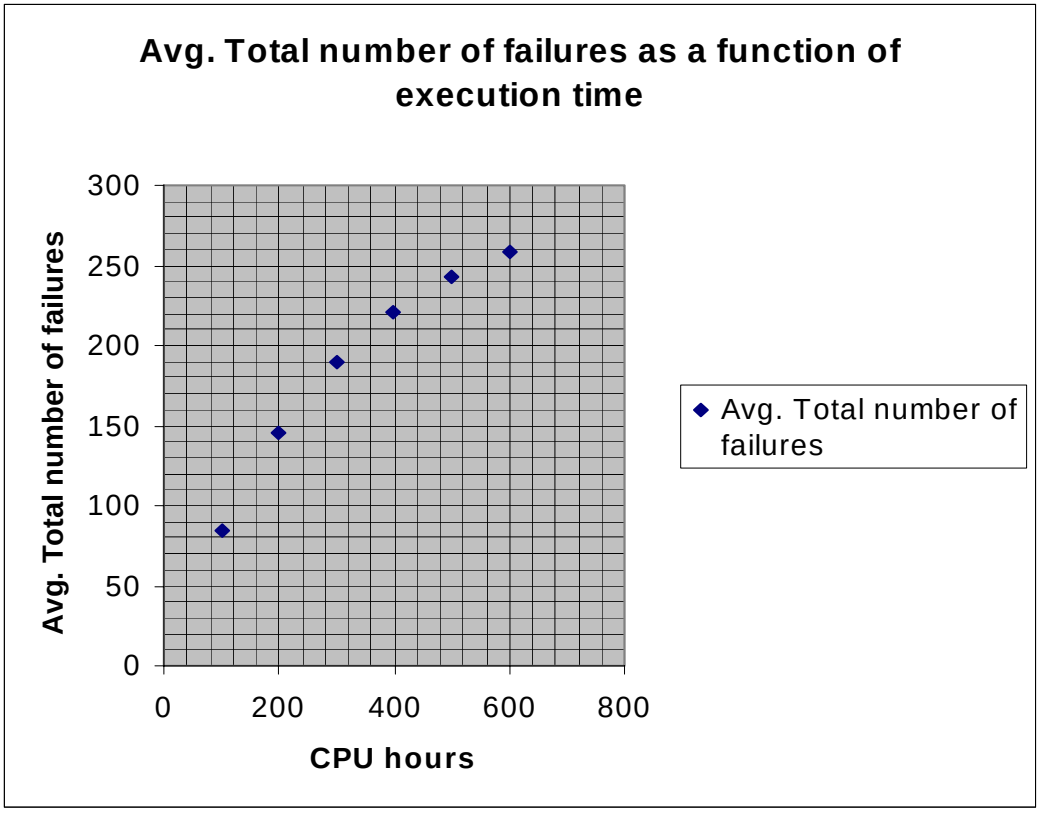
4. If one wanted to create a generic GUI application that may be used not only to convert Celsius to Fahrenheit temperature values but to do arbitrary conversions (e.g. centimeters to inches, kilograms to pounds, etc.) what additional design pattern would be most appropriate for achieving this flexibility, and how will it be used? Please show the enhanced design by giving a class diagram.

Q6. Reliability and Testing

1. Define the terms failure, error, and fault

2. What are the basic assumptions of Musa's software reliability models

3. The figure below shows the average total number of counted failures as a function of the execution time (in CPU hours). From past experience with this type of product, we know that the product is released when its testing failure rate is 0.005/CPU hour, and that the estimated total number of failures, (v_0) that can appear over infinite time is 300. Using the model, determine the total estimated testing time (t) in CPU hours so that the software can be released, and the estimated average total number $\mu(t)$ of counted failures at the time of its release. How many additional hours do we need to test to achieve a field failure rate 0.0001? (Note ln(0.0001) = -9.21).
You can consider using the Musa's equations pertaining to average total number of failures as a function of time $\mu(\tau)$ and the failure rate as a function of time $\lambda(\tau)$.



Consider that the basic model is

$\mu(\tau) = v_0 (1 - \exp(- \lambda_0 \tau / v_0))$ and

$\lambda(\tau) = \lambda_0 \exp(-\lambda_0 \tau / v_0)$

Useful Numbers

Number	LN	Number	EXP
0.003	-5.80914	-5	0.006738
0.004	-5.52146	-5.1	0.006097
0.005	-5.29832	-5.2	0.005517
0.006	-5.116	-5.3	0.004992
0.007	-4.96185	-5.4	0.004517
0.008	-4.82831	-5.5	0.004087
0.009	-4.71053	-5.6	0.003698
0.01	-4.60517	-5.7	0.003346
		-5.8	0.003028
0.09	-2.40795	-5.9	0.002739
0.1	-2.30259	-6	0.002479
0.11	-2.20727		
0.12	-2.12026		
0.13	-2.04022		
0.14	-1.96611		
0.15	-1.89712		