Chapter 12

Exercises and Project Suggestions

EXERCISE 1:

USA Today on December 31, 1998 carried an interesting article, "Birth of a New Order", talking about the year that world's lines of time and space collapsed. The year is of course 1998. The most incisive paragraphs are excerpted below:

The global, time-crunched market driven by electronic information "forces things to get bigger and smaller at the same time," says Nicholas Negroponte, author and technologist at the Massachusetts Institute of Technology. "And that's so ironic, when things want to do both but not stay in the middle. There will be an increasing absence of things that aren't either very local or very global". Oil and cars aren't much suited to being small and local. So they're moving to become gigantic and cross-border.

As for being small and local, that's where the Internet, or World Wide Web, comes in -- and it works in two ways. It lets little companies be global, so a start-up in a garage can put its goods or services on a Web site and sell worldwide, competing against midsize or big companies, wiping out disadvantages (such as distribution and scope) that once had to do with distance. And since little companies can change direction faster than bigger ones, they have an advantage in time. Big companies used to have time and distance on their side. Increasingly, little ones do.

And so in 1998, we had the phenomenon of Amazon.com, which has become such a symbol of small beating big that business people have turned it into a verb: to be "amazoned".

.

In the context of the above, write a mini-essay to discuss what do you envision multimedia software engineering will become, and how multimedia software engineering might help the "little guys" compete against the "big guys", or the other way around. The mini-essay should be between 1,500 and 2,000 words, with no less than 3 and no more than 5 references. A "template" is provided:

1st paragraph: My vision of multimedia software engineering in the year 200X.
2nd paragraph: My vision of a small company in the year 200X.
3rd paragraph: A scenario of the small company in action.
4th paragraph: Reasons why MSE can help the small company compete against the big guys.
5th paragraph: More discussions.
3 to 5 references.

The mini-essay should be e-mailed to the instructor by the deadline.

EXERCISE 2:

The purpose of this exercise is to enable the students to gain familiarity with the active index approach to active information system design. As discussed in the book, the hypermedia model and the active index together can be used to model active distributed multimedia information systems. In this exercise we will first concentrate on the active index component.

Let us consider an adaptive distance learning system.

The distance learning materials are organized into a hypermedia structure. A student user can browse through these multimedia documents and follow the links to access related multimedia documents. As such, the hypermedia structure is passive, waiting to be accessed by the user.

We can make the hypermedia structure active by associating index cells with selected multimedia documents. The idea is to designate a special document so that when many students access this document, it means they have reached a certain level of proficiency and therefore the learning materials should be adjusted to become more difficult. Likewise when many students access a special document indicating deficiency, it means they have problems and therefore the learning materials should be made simpler.

The following index cell types are specified:

Proficiency-level index cell: The proficiency-level index cell is associated with a certain specific multimedia document (such as doc-1, usually reacheable only by proficient students). When this index cell is

triggered, it will increase the proficiency-level by 1. When the proficiency-level has reached a predefined threshold (such as 3), it will send message to the instructor, informing the instructor that a sufficient number of students have reached this level of proficiency. It will also send messages to certain documents (such as doc-3, doc-4, doc-5) to become harder.

Deficiency-level index cell: The deficiency-level index cell is associated with a certain specific multimedia document (such as doc-2, usually reacheable only by deficient students). When this index cell is triggered, it will increase the deficiency-level by 1. When the deficiency-level has reached a predefined threshold (such as 2), it will send message to the instructor, informing the instructor that a significant number of students have reached this level of deficiency. It will also send messages to certain documents (such as doc-3, doc-4, doc-5) to become easier.

Self-adjustment index cell: This self-adjustment index cell is associated with multimedia documents containing learning materials (such as doc-3, doc-4 and doc-5). When it receives a "harder" message, it upgrades the learning materials to become harder. Likewise, when it receives a "easier" message, it downgrades the learning materials to become easier.

The above are three index cell types. The instances can be associated with individual multimedia documents (such as doc-1, ..., doc-5).

There is also a home page (such as doc-0), with links to the other documents (such as doc-1, ..., doc-5).

(a) Draw state-transition diagrams to define graphically the three index cell types.

(b) Specify the three index cell types formally using mathematical notations ic = (X, Y, S, s_o, A, t_{max}, f, g).

(c) Draw a diagram showing a few multimedia documents (such as doc-1, ..., doc-5) enhanced with the index cells to illustrate how these index cells work together to form an active index system.

(d) Use the IC Builder to construct the three index cell types. The output from IC Builder, together with the appropriate actions (C functions) and specification of input message space, output message space, will become input to the IC Compiler to generate the IC Manager.

How to download the IC Builder:

Please go to the author's web site at: www.cs.pitt.edu/~chang and follow the links to multimedia software engineering courseware. In a directory for IC_Builder the following files can be found: ictapp.zip and ictype.zip. Use pkunzip to unzip and install under Windows.

At the time of the writing of this book, the above mentioned files are at:
http://www.cs.pitt.edu/~jung/IC_Builder/ictapp.zip
http://www.cs.pitt.edu/~jung/IC_Builder/ictype.zip

Additional Explanation:

For this exercise there is NO NEED to write any C functions. The assignment can be handed in either as a hard copy or via the Internet. For the part where you use IC_Builder to construct IC types, you can turn in the output file(s) generated by the IC_Builder, which are ascii files X.in. You can also provide screen dumps captured during the construction process. If you use Internet, it will be the best if you can provide the URL so that the instructor can browse the web pages containing the solutions. In other words, please prepare a set of web pages and figures can be embedded as gif/jpg files. This will be the easiest for other people to read. It will also be useful when you later develop a presentation based upon such materials.

EXERCISE 3:

The purpose of this exercise is to understand the relationship between active index and Petri nets. Both are tools for the modeling of distributed multimedia systems. Active index cells are added incrementally to build a dynamic index, and the connections can also change dynamically. However, if the massages passed between index cells are deterministically routed, then it is possible to convert active index into a Petri net. Otherwise you must use a Petri net with conditions (predicates) associated with the transitions, or an Evaluation Net (E-net).

(a) Convert the active index you constructed in Exercise #2 into a Petri net (or an E-net).

(b) Take the diagram you drew in part (c) of Exercise #2. Redraw it here (because you may want to make some changes), and now use the marked Petri net to illustrate the scenario. You can draw a sequence of marked Petri net to show how the system works.

Additional Explanation:

(a) If we consider how the active index system passes messages and reaches equilibrium state (if one exists), this leads to a formal study using, for instance, the Petri net model.

(b) Notice this is the beginning of a systematical approach to build prototypes for active distributed multimedia systems. Can we create a new systematic approach, i.e., a new software process model, for distributed multimedia systems design?

(c) The index cells could span several nodes. Therefore, the active index system is a distributed index. The IC Managers must also be distributed to the nodes in the networks.

EXERCISE 4:

The purpose of this exercise is to experiment with MICE, the prototyping tool for distributed multimedia computing. The MICE development environment provides step-by-step instructions on how to use the IC_Builder to create the Ics, how to use the IC_Compiler to create the customized IC_Manager, and finally how to generate the multimedia application.

First, you need to compile the index cell specifications using the IC_Compiler explained in Section 3 of Chapter 8. The IC_Compiler is used to create the customized IC_Manager. This customized IC_Manager explained in Section 4 of Chapter 8 becomes the CGI program to be invoked when the user clicks on the Web pages.

Then, you build the HTML pages for the active index system for distance learning you did for exercise #2. The end result should be a demonstration. You only need to e-mail your URL to the instructor so that the instructor can try your demo.

After that you can set up your working directory, by creating two subdirectories called "TAOML" and "source", and then copying all the files from the three directories IC_Compiler, IC_Manager and IC_Taoml to this "source" directory. Then you can follow the steps spelled out in MICE Application Development Steps in Section 6 of Chapter 8.

All the necessary files are available under ~jung/public/html in the following four directories: IC_Builder (the files you need to run the IC_Builder on PC), IC_Compiler (the files to run the IC_Compiler), IC_Manager (the files needed to compile the IC_Manager) and IC_Taoml (the interpretor to translate .taoml pages to .html pages) and can be downloaded.

EXERCISE 5:

After you have studied the various approaches, write a critique of these approaches, and a proposal of what you intend to do as a project. The critique and proposal should be between 1,500 and 2,000 words, with no less than 5 and no more than 10 references. A "template" is provided:

1st paragraph: Introduction.

2nd paragraph: My critique of various approaches.
(this could be the longest paragraph)
3rd paragraph: What I propose to do and how.
4th paragraph: Why this project is worth doing.
5th paragraph: Discussions.
5 to 10 references.

The critique/proposal should be e-mailed to the instructor.

PROJECT SUGGESTIONS:

Select an application area of interest to you. Design a multimedia application according to the suggested framework:

1. Syntax: Design the web pages and multimedia interaction languages.
2. Semantics: Design the actions associated with an application, and develop the active index cells (or agents). Associate the syntax with the semantics to organize the tele-action objects.
3. Pragmatics: Prototype the application using the MICE prototyping and multimedia information custom engineering environment.
4. Critique: Evaluate your design and suggest different design alternatives.

Those who are more interested in theory can investigate the formal specification of a multimedia application, and the verification and validation of such a formal specification.