SWENG 587 Final Examination

Instructions

Max Points: 30

Please answer all questions that follow based on the alternative designs described for the

text processing system. The answers should be clear and to the point. Good luck!

Consider a system that accepts an ordered set of lines; each line is an ordered set of words; and

each word is an ordered set of characters. Any line may be “circularly shifted” by repeatedly

removing the first word and appending it at the end of the line. The system outputs a listing of all

circular shifts of all lines in alphabetic order. For example:

Input: ------------answer the following question

-------------------it should be clear

Output:-----------be clear it should

-------------------clear it should be

-------------------following question answer the

-------------------question answer the following

-------------------should be clear it

-------------------the following question answer

There are three different architectural solutions to this problem. Figure 1 shows the first solution.

Figure 1: Solution 1 to the problem

In solution 1 to the problem:

• There are four basic functions – input, circular shift, alphabetize and output

• They are coordinated as subroutines by a main program – master control

• Data is communicated between the components through a shared storage (i.e.,

characters, index, and alphabetized index are in a shared storage that all functions

can access)

Please Note: In solution 1, lines are processed simply as a sequence of characters;

there is no abstract notion of a line. So the Input module stores each line as a

sequence of characters, the Circular Shift module simply stores an index to various

characters in the line that mark each possible circular shift of that line, and the

Alphabetizer module maintains sorted order of those indices (essentially

representing circular shifts of a line in sorted order). So all modules essentially look

at the same copy of a line (a sequence of characters), at the same copy of indices (a

circular shift of the lines) and at the same copy of the alphabetized index (a sorted

circular shift of the lines).

Figure 2 shows the second solution.

Figure 2: Solution 2 to the problem

Solution 2 is similar to the first solution with two important differences:

• First, the interface to the data is more abstract. Rather than exposing the storage

formats to the computing modules, data is accessed abstractly (for example, as a list

or a set).

• Second, computations are invoked implicitly as data is modified. Thus interaction is

based on an active data model. For example, the act of adding a new line to the line

storage causes an event to be sent to the shift module. This allows it to produce

circular shifts (in a separate abstract shared data store). This in turn causes the

alphabetizer to be implicitly invoked so that it can alphabetize the lines.

Please Note: As the legend on second solution shows, the arrows on the diagram

represent two kinds of dependencies or relationships. The non-­‐filled arrowhead is

for implicit invocation (so the line module implicitly invokes the circular shift

module which implicitly invokes the alphabetizer module). The filled arrowheads

show the sequence of calls – Master Control module calls Input module which calls

insert operation of the Lines module (a data store for lines) which leads to implicit

invocation of the i-­‐th (or read the i-­‐th line) operation of the Circular Shift module

and the corresponding writing of this line using the insert operation of the Lines

module (a separate data store which keeps the lines in sorted order). This write to

the separate data store implicitly invokes the Alphabetizer module which reads the

i-­‐th line and inserts it back into the ate store in its proper sorted order. Finally,

when all input has been processed, the output module is called by the Main Control

module to display the sorted lines.

Figure 3 shows the third solution.

Figure 3: Solution 3 to the problem

In solution 3 to the problem:

• There are four filters: input, circular shift, alphabetize, and output.

• Each filter processes the data and sends it to the next filter.

• Control is distributed: each filter can run whenever it has data on which to compute.

• Data sharing between filters is strictly limited to that transmitted on pipes.

Questions

Please give a brief summary providing evidence for strengths or weaknesses of the

proposed architectures with respect to the following:

(1) Ease of changes in the processing algorithms: For example, line shifting can be performed on

each line as it is read from the input device, on all the lines after they have been read, or on

demand when the alphabetization requires a new set of shifted lines. (10 Points)

(2) Ease of changes in the data representation: For example, lines, words and characters can be

stored in various ways. Similarly, circular shifts can be stored explicitly or implicitly. (10

Points)

(3) Performance: Both space and time. (10 Points)