**CSCI 5300, Spring 2020. Instructor: Phil Pfeiffer**

**Assignment 04: ISC assignments, part 3: design considerations for architectural information flows**

**Value: 30 points - 15 problems, 2 points each**

**Terms: as individuals**

**Due date: Saturday, March 7, 2020**

***Assignment***. A description of a system’s architecture should account for flows of content into the system and amongst its components. This assignment asks you to reason about the flow-related concerns for the postulated International Storytelling Center system given below.

* ***Flow of content into the system from external sources.*** When creating an architecture to constrain what inputs a system will accept and where and how those constraints will be applied.

*Problems:*

1. What difficulties could be posed by allowing users to register for accounts? What measures could be taken to address these difficulties? What component(s) should implement these measures?
2. What difficulties could be posed by allowing users to specify the characteristics of the tells about which they wish to be notified? What measures could be taken to address these difficulties? What component(s) should implement these measures?
3. What difficulties could be posed by the uploading of tells from the tell hosts? What measures could be taken to address these difficulties? What component(s) should implement these measures?
4. What difficulties could be posed by the importing of schedules from the ISC's festival database? What measures could be taken to address these difficulties? What component(s) should implement these measures?

* ***Flow of tells from tell host controllers to the tell event controller.*** I see three ways in which this flow could be managed:
* *Pull from the tell event controller, via the venue collection****.*** Here,
  + the tell event controller polls tell host controller for tells, via the venue collection.
  + uploading tells when they become available
* *Push from tell host controllers, via the venue collection.* Here,
  + tell host controllers
    - communicate with the venue collection when a tell becomes available,
    - uploading that tell to the venue collection
  + the venue collection, in turn, uploads the tell to the tell host controller
* *Providing support for pull from the tell event controller as well as push from tell host controllers.*

*Problems:*

1. For each of these three possibilities, describe that possibility's primary advantages--or, if you believe that it offers no advantages, its primary disadvantages.
2. Specify which of the three options you favor, and why. Then, update the accompanying developer diagram as needed to account for your strategy of choice.
3. Based on your answers to the previous two questions, create an activity diagram that shows the flow of activity that results in the acquisition of a tell by the tell event controller. Be sure that your diagram accounts for the following:

* the action or actions that put the process in motion
* the action or actions that end the process
* actions that the system could take along every step of the process, should the preferred action fail

1. *Multithreading*[[1]](#footnote-1) is a strategy for improving a system's performance by allowing multiple computations to run concurrently--and, ideally, in parallel, if the hardware supports the simultaneous execution of multiple computations. Multithreading is often implemented using *thread pools*[[2]](#footnote-2): fixed-size collections of reusable "mini-processes" that are allocated to requests for service.

Can activity diagrams be used to represent the operation of multithreaded computations like the ones that might be implemented to handle concurrent uploads of tells? If so, explain how. If not, discuss the difficulties that preclude such a representation.

* ***Flow of tells from tell host controllers to the tell event controller (cont.)***  I can envision two paths along which tells could flow from tell host controllers to the tell event controller:
* *Via the venue collection,* as shown in the accompanying developer diagram*.*
* *Directly through the tell host interfacer,* using the venue controller as a resource that informs the communication*.*

*Problems:*

1. For each of these two possibilities, describe that possibility's primary advantages--or, if you believe that it offers no advantages, its primary disadvantages.
2. Specify which of the two options you favor, and why. Then, update the accompanying developer diagram as needed to account for your strategy of choice.

* ***Flow of tells from the tell event controller to the tell collection.*** I see three ways in which this flow could be managed:
* ***Push from the tell event controller, which transfers the tell directly to the tell collection***. Here,
  + The tell event controller, at the point where it starts to upload a tell, obtains a reference to a routine for storing this tell's content from the tell collection.
  + The tell event controller then uses this referenced routine to store the tell.
* ***Push from the tell event controller, which directly or indirectly informs the tell collection that an information holder can be consulted to learn about the new tell***. Here,
  + The tell event controller,
    - at the point where it starts to upload a tell, obtains a reference to a routine for storing this tell's content from an external information holder
    - uses this referenced routine to store the tell.
    - either
      * informs the tell collection that the tell has been uploaded, leaving it to the tell collection to consult the information holder about the change, or
      * relies on the information holder to inform the collection of the change
* ***Pull by the tell collection, which retrieves the tell from the tell event controller***. Here,
  + The tell event controller, upon receiving a tell, stores the tell in an internal cache of tells.
  + The tell collection, acting independently of the tell event controller at some later time,
    - polls the tell event controller, moving all tells that the tell event controller has buffered since it was last polled into the tell collection.
* ***Pull by the tell collection, which retrieves the tell from an information holder***. Here,
  + The tell event controller, upon receiving a tell, stores the tell in an information holder (i.e., tell holder).
  + The tell collection, acting independently of the tell event controller at some later time,
    - polls the information holder, moving all tells that the holder has buffered since it was last polled into the tell collection.

*Problems:*

1. For each of these four possibilities, describe that possibility's primary advantages--or, if you believe that it offers no advantages, its primary disadvantages.
2. Specify which of the four options you favor, and why. Then, update the accompanying developer diagram as needed to account for your strategy of choice.

* ***Flow of information from the tell collection to the notification service.*** I see four ways in which this flow could be triggered:
* ***Push from the tell collection, which provides the service with information about updates to the collection.*** Here,
  + The tell collection, upon receiving a tell or change in tell status,
    - updates its collection of tells, then
    - invokes the tell notification service, passing information about the update.
* ***Push from the tell collection, which directly or indirectly informs the service that an information holder can be consulted to learn updates to the collection***. Here,
  + The tell collection, upon receiving a tell or a change in tell status,
    - updates its collection of tells;
    - writes an update notification to an external information holder; then
    - either
      * informs the tell notification service that the collection has been updated, leaving it to the service to consult the information holder about the update, or
      * relies on the information holder to inform the service of the change
* ***Pull by the notification service, which retrieves updates from the tell collection.*** Here,
  + The tell collection, upon receiving a tell or change in tell status, updates its collection of tells.
  + The notification service, acting independently of the collection at some later time,
    - polls the tell collection service, retrieving information about all updates that the tell collection has received since the notification service last polled the collection.
* ***Pull by the notification service, which retrieves updates from an information holder.*** Here,
  + The tell collection, upon receiving a tell or change in tell status,
    - updates its collection of tells and
    - writes an update notification to an external information holder.
  + The notification service, acting independently of the collection at some later time,
    - polls the information holder, retrieving information about all updates about which the information holder has been notified since the notification service last polled the collection.

*Problems:*

1. For each of these four possibilities, describe that possibility's primary advantages--or, if you believe that it offers no advantages, its primary disadvantages.
2. Specify which of the four options you favor, and why. Then, update the accompanying developer diagram as needed to account for your strategy of choice.
3. Based on your answers to the previous two questions, create a sequence diagram that shows the flow of requests among the participating entities. Be sure that your diagram shows

* the entity or entities that put the process in motion and
* the entity or entities that the notification service targets.

***Deliverables.***

* A Word document with answers to questions 1-6 and 8-14.
* Diagrams for questions 7 and 15.
* An updated version of the accompanying developer diagram, according to the instructions in questions 6, 10, 12, and 14.

1. see <https://en.wikipedia.org/wiki/Multithreading_(computer_architecture)> [↑](#footnote-ref-1)
2. see <https://en.wikipedia.org/wiki/Thread_pool> [↑](#footnote-ref-2)