HW6 – Design Communication for Distributed Systems with Vitruvian

Due Date: *12/7/2012*
Estimated time: *1-2 hours*

Objectives

- Begin think about how to design communications for distributions systems, in terms of object synchronous patterns.

Overview

For this homework assignment, you will review the attached analysis and design diagrams, as well as the source code for the Ants Simulation. The goal is to determine what kinds of object synchronization will be most effective for each method that needs to support remote access.

The possible synchronization patterns are as follows:

Constant	This would be better named "Once only". The proxy object retrieves the results of the method or property from the host object just once, and then caches it local for all further uses. If the host object's state changes afterward, then the proxy object will never see that change.
RPC	This stands for *Remote Procedure Call*, but would be better called *Remote Method Invocation*. For every invocation of a method or property on a proxy object, the proxy object requests that method or property to be executed on the host object and receives the results back in a reply. This is a synchronous synchronization pattern.
APRC	This stands for *Asynchronous Remote Procedure Call*. For every invocation of the method or property on a proxy object, the proxy object requests the method or property to be executed on the host object, but the holder of the proxy object can continue doing whatever it needs to do. Later, it can check or wait for the request method or property to finish. Obviously, this is asynchronous synchronization pattern.
Push	If no time frequency is specified, then this pattern triggers synchronization whenever the state of a property changes. The synchronization is a one-way push of the property's state to all other instance of the distributed object. This pattern cannot be used on a method without a frequency. When a time frequency is specified, then the sync pattern checks periodically to see if the

	state of change object has changed. Then, if it has, it sends the new state to all other instances of the distributed object.
Pull	If this pattern has no frequency, then it is the same as the RPC pattern. With a frequency, the sync pattern will periodically do an RPC-like retrieve of the host object's state. The value return by the reply is cached locally and returned in response to local invocations of the property or method.
Host Only	This pattern restricts the method so it will only execute on the host object, even though it is a public method and, therefore, visible to proxy objects.
SmartList	This is a special pattern for optimizing the synchronization of a list of objects.

Instructions

After you review the diagrams and code, and make some design decisions, complete the attached HW6-DesignChoices.docx documents.

Submission Instructions

Submit the HW6-DesignDecisions.docx documentation for HW6 on Canvas.

Grading Criteria

Obvious lack of effort	0 – 49
Incomplete	50 – 69
Reasonable design chooses, but less than convincing justifications	70 – 79
Good design chooses and good justifications	80 – 89
Excellent design chooses and excellent justifications	90 – 99