Dejan Ahmetovic

9/16/2016

CS 6310 Software Architecture and Design

Assignment 5: Course Management System – Next Design Phase (v3)

Architecture Styles

The three architecture styles I have chosen for the problem are: Repository, Data Abstraction and Object-Oriented Organization and Client-Server.

Repository architecture style fits good with the problem here because the types of transaction in an input stream of transactions trigger selection of processes to execute, the repository can be a traditional database. When I think of a repository I think of a database. This feature of repository will play a nice role here since we have to store the data somewhere where it can be accessed by anyone and anywhere. We have many students across the world that would want to take the courses so they have to be able to see what the program has to offer. This architecture provides data integrity, backup and restore features. Also this style provides scalability and reusability. Moreover, this style is also platform independent and a tradition database is one of the main concepts that uses CRUD (Create – Retrieve – Update – Delete). Since our admins must have this ability this architecture will fit right in with the problem.

The other good and usable architecture that can be applied to our problem is Data Abstraction and Object-Oriented Organization. In this style the objects are modeled as the real entities in the world. Representation is hidden from other objects which promotes encapsulation where data is hidden and only belongs to a specific entity. Data Abstraction and Object-Oriented

Organization style also has the best ability to decompose problems into collections of interacting agents which help solve the problem in smaller pieces.

My last choice for this problem was the Client – Server architecture style. In this style we have many clients that can connect to a server from anywhere as long as there is a network connection. Clients issue requests to a server and get some responses back. Each client is independent and there can be many clients at the same time accessing servers. Servers need to be able to accept and process these requests and send the results to the clients. Since all the data is stored at the same place, data management becomes easy.