Dynamic Presenter – UI & Database Controller Subsystem

Student: Jimmy Mauri, Florida International University
Mentor: Juan Caraballo, IBM
Instructor: Masoud Sadjadi, Florida International University

Problem

General Problem:
A company may show presentations to an audience for a multitude of reasons. Making a presentation is time consuming work, so multiple collaborators may want to work on the same presentation to speed up development. However, in collaborative projects it is difficult for managers to effectively control the changes made by other contributors and to keep track of multiple versions of the same presentation.

Specific Problem:
The system ultimately requires other users to be able to interact with the system directly and with each other. In addition, the system needs to be able to control authorization such that it can manage what users can manipulate which projects and to which extent.

Current Systems

Advantages:
• Allow multiple users to work on one presentation at the same time.
• Have version histories that allow users to keep track of the changes made to the presentation at a project-wide level.

Drawbacks:
• Any contributor can make changes without the approval of the presentation creator and without the knowledge of all other contributors.
• Version history does not keep track of changes at the slide level.
• Users have no way of blocking other contributors from seeing the contents of specific slides.

Requirements

User Interface Subsystem General Requirements:
• Act as an intermediary between other subsystems and the client.
• Handle http requests from the client.
• Communicate with the DB Controller for authorization on projects.
• Communicate with VCS for presentation information.
• Handle session information between users.

Database Controller Subsystem General Requirements:
• Handle user authorization at project level.
• Register users to system.
• Manage addition and removal of users in a project.
• Manage user roles for each project.
• Manage addition of projects.
• Return information related to projects and individual roles in each project.

System Design

DP - Architecture Diagram:
DP has a client-server architecture. UI depends on the VCS, and the VCS depends on the Parser.

DP - Deployment Diagram:
FileSystem and DBMS can be deployed on dedicated servers if necessary.

Implementation

• Implemented in Python.
• Uses Flask Web Framework to handle page requests from the client to the user interface.
• Utilizes Jinja to render web templates, customizing each template to be relevant for each user and each project in accordance to their role.
• Database component uses psycopg2 to maintain, modify and create a database for appropriate authorization and role usage.
• Psycopg2 utilizes PostgreSQL to perform tasks.
• Flask handles session management for the system.

Screenshots

• User Interface uses Flask web framework to facilitate outside interaction with system.
• DBC handles authorization requests for system.
• Subsystem allows project managers to dictate what gets authored in each presentation.
• Subsystem allows for many contributors to individually submit slides for each presentation.
• System allows for presentation creators to build presentations for all contributors based on slide creator’s contributions.
• Subsystem enforces role for each project so that users in specific roles cannot modify the project beyond their designated role.

Object Design

Solution to general problem:
Dynamic Presenter: A Presentation Management System that allows multiple users to work on one presentation simultaneously. Presentation creators have complete control over the changes made by other collaborators. An extensive version history allows users to revert an entire presentation to a previous version, as well as individual slides without affecting the rest of the presentation.

Solution to specific problem:
UI Component: Provide a means to visually represent the information existing in the system in a usable manner for users. Communicate with the DBC component and VCS subsystem.

DBC Component: Check authorization rules for all users and all projects.

Summary

I would like to recognize the hard work and dedication put in by my group members Carlos Fernandez (Version Control System) and Jose Camino (Parser) throughout the duration of this project.