Team Crystal Project:

Aggregation and Archiving of Artifacts

Joshua Harris

CS410

Old Dominion University

May 4, 2020

Introduction

Currently schools lack a framework with the ability to aggregate and archive artifacts. Artifact repositories are locked behind individual classes and degrees. This means that students are limited when it comes to expanding their knowledge in additional subjects. Faculty members are valuable unique resources to universities. Over the course of their tenure faculty members gather valuable materials that they share with their students. This is an amazing asset for both students and other faculty members. Teachers move on and retire and more often than not their collected resources vanish along with them. There is currently no way for teachers to pool their resources together properly. Current aggregation methods are not strong enough to be considered centralized. These systems also tend to lack the ability to track changes to artifacts over time. What these current aggregation systems also lack is uniformity when it comes to their resources. The formats of artifacts can vary rapidly which can create confusion among the students.

A Cube Solves these issues with a wide variety of algorithms and features. A cube is a framework for aggregating and archiving artifacts for educators, researchers, and students. A Cube will allow faculty members to store and update class resources with an easy to use user interface. The system will track changes made to artifacts by repository owners. This will keep students informed as to what resources are up to date. Reports can also be generated to provide information to system administrators about users as well as permissions. A cube is a centralized database that allows students to access repositories without class enrollment restrictions.

A cubes primary feature is the normalization of uploaded artifacts. Uploaded artifacts are converted to markdown; this will become important when we discuss our additional features.

Updating artifacts manually or through web scraping will automatically trigger a comparison or Diff with the original file. All repositories and artifacts will have specific tags attached to them. These tags are used for a number of different reasons that I will discuss later in the paper.

Product Description

Put more simply A cube is a large database that houses faculty resources. These resources can be accessed from anywhere by any user with the appropriate permissions. Faculty members and above will be able to create their own repositories in which they can store appropriate files. As mentioned above each artifact and repository will have tags attached to them. These tags will allow users to search for specific artifacts based on the tags themselves or the dates uploaded. Users above the guest level will also be able to bookmark specific repositories to access conveniently. Contributors and followers of each repository will be notified of all changes that occur to artifacts.

Users of different permissions levels will be connected to a centralized repository. Key features of A cube will allow faculty members to create their own repositories to store artifacts. Each artifact and repository will be linked to user defined tags. These tags will be used for user searches. Searches can be done based on tags and last updated. A cube will give users the ability to compare the differences between artifacts. This is accomplished using normalization. Uploaded artifacts will be converted to markdown format allowing for line by line comparisons. These differences will then be displayed to the user via a user interface. A cubes notification algorithm will notify users of any changes made to repositories. A cube also utilizes web scraping allowing it to update and upload artifacts to the database.

A cube will work off the ODU severs. Users will be able to access the database from any web capable computer. A cube will be coded in Python 3.8 or newer. Our user interface will be made utilizing HTML, CSS and Javascript. Javascript frameworks to be used are angular and react. The respective teams will be utilizing Visual Studio Code as the IDE for development. Pydoc and Sphinx will be used for documentation purposes. The teams will use GitLab to share and update code. Containerization and deployment will be handled by Docker. The database has not been decided on and the choices have been narrowed down to either MySQL or MongoDB. The application programming interface will be a Restful API. Analysis portion of A cube will be handled in Pydocstyle.

Identification of Case Study

A cube is the brainchild of ODU computer scientist instructors. The computer science department felt that they could benefit from a centralized database of faculty resources and thus A cube was born. The system will be used for the aggregation of artifacts by teachers. Students will have access to all public and private repositories as long as the appropriate permissions are granted by the owner. ODU is only the beginning, all of academia could benefit from A cube. Other colleges could benefit from the program.

Glossary

(Guidelines - Alphabetical list of terms and abbreviations.)

Aggregate: Data that is composed of smaller pieces that form a larger whole.

Algorithm: Set of instructions designed to perform a specific task.

Angular: A framework for dynamic web apps. Allows for the use of HTML as a template language.

Application Programming Interface (API): Set of functions and procedures allowing the creation of applications that access features of an operating system, applications, etc.

Archive: Contains multiple files and/or folders. May be created by several different utilities and may be saved in different formats.

Artifact: Combination of arte, “by skill”, and factum, “to make”. A file or document.

Backlink: A hyperlink that links from a web page, back to your own web page or website.

Blackboard: A tool that allows faculty to add resources for students to access online.

Centralized: Type of network where all users connect to a central server.

Course Websites from Markdown (CoWeM): A system for building course websites, including notes, slides, and organizational pages, from Markdown documents.

Cascading Style Sheet (CSS): Used to format the layout of web pages. Defines text styles, table sizes, among other things that previously could only be defined in HTML.

Database: Collection of information, that is organized for rapid search and retrieval.

Data Loss: An instance in which information is destroyed by failures or neglect.

Diff: A line by line comparison of normalized artifacts.

Docker: Tool to create, deploy, and run applications by using containers. Allow developers to package up an application, with all parts needed, to be deployed in one package.

Export: Taking data from one program or computer to another.

GitLab: Used to provide internal management of git repositories. Is a self hosted Git-repository management system that keeps the user code private.

Graphical User Interface (GUI): User interface that contains graphical elements. Examples

include windows, icons and buttons.

Hypertext Markup Language (HTML): A language used to create web pages. “Hypertext” refers to hyperlinks in a page, and “Markup language” refers to the way tags are used to define page layout.

Hyperlink: An element that links to another file or object.

JavaScript (JS): A language used in web development. While influenced by Java, It’s syntax is more similar to C.

Knowledge Management: The management process of creating, capturing, sharing, retrieving, and storing data, information, knowledge experiences and skills by using appropriate information and network technology.

Markdown: A markup language that can be used to format plain text. Can be converted into another language.

Markup: A language that uses tags to define elements within a document.

MySQL: Open source SQL database management system. Developed and distributed by Oracle Corporation.

Normalization: Converting ingested objects into a small number of pre-selected formats.

Python: An interpreted, object-oriented language.

Personal Learning Environment (PLE): An interface used in flexible online courses. Designed by ODU’s Center for Learning and Teaching.

pydoc: Automatically generates documentation from Python modules. Can be presented as pages of text on the console, served to a web browser, or saved to HTML files.

Pylint: A Python static code analysis tool. Looks for programming errors and warnings from within the code, as well as from an extensive configuration file.

React: A JavaScript library that is used to create User Interfaces for web applications.

reStructuredText: A plaintext markup syntax and parser system. Useful for in-line program documentation.

Secure File Transfer Protocol (SFTP): Secure version of File Transfer Protocol. Facilitates data access and data transfer over a Secure Shell data stream

Sphinx: A Python documentation generator. Converts reStructuredText files into HTML websites and other formats.

Tags: Is a keyword or term assigned to a piece of information.

tox: Aims to automate and standardize testing in Python. Is a generic virtualenv management and test command line tool.

Visual Studio Code: A source code editor that runs on Mac, Linux, and Windows.

2. References

(Guidelines - Listing of sources.)

Blackboard Archive Extractor. (2016, December 15) cs.odu.edu. Retrieved March 10, 2020, from https://www.cs.odu.edu/~cpi/old/411/crystals17/.

Carroll, J., Choo, C. W., Dunlap, D., Isenhour, P., Kerr, S., MacLean, A., & Rosson, M. (2003). Knowledge Management Support for Teachers. *Educational Technology Research and Development*, *51*(4), 42-64. www.jstor.org/stable/30221184

Davenport, T., Long, M. & Beers, M.. (1997). *Building Successful Knowledge Management Projects* [Working Paper]. Retrieved March 8, 2020, from https://www.researchgate.net/publication/200045855_Building_Successful_Knowledge_Management_Projects.

Document Management Software | eFileCabinet. (2020). eFileCabinet. Retrieved February 20, 2020, from https://www.efilecabinet.com.

Domes, S. (2017). *Progressive Web Apps with React: Create lightning fast web apps with native power using React and Firebase.* Packt Publishing Ltd.

File Sharing and Sync For Education, Schools and Universities - FileCloud. (2020). FileCloud. Retrieved February 20, 2020, from https://www.getfilecloud.com/file-sharing-and-sync-for-education/.

GitHub Features: The right tools for the job. (2020). GitHub. Retrieved March 10,il

2020, from https://github.com/features#team-management.

Kennedy, T. (2020, January 21). *Home · Wiki · Thomas J. Kennedy / cs-roars-proposal*. GitLab. Retrieved 26 April 2020, from https://git-community.cs.odu.edu/tkennedy/cs-roars-proposal/-/wikis/home.

Nvlpubs.nist.gov. (n.d.). *Glossary of Key Information Security Terms.* From https://nvlpubs.nist.gov/nistpubs/ir/2013/NIST.IR.7298r2.pdf.

MacFarlane, J. (2006). *Pandoc - About pandoc*. Pandoc.org. From https://pandoc.org/index.html.

Tsapps.nist.gov. (2020). *Data Loss Prevention.* From https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=904672.

Xie, I., & Matusiak, K. K. (2016, July 29). Digital preservation. *Science Direct* (255-279). Retrieved March 10, 2020, from https://www.sciencedirect.com/science/article/pii/B9780124171121000090

Zeil, S. (2019, December 26). *Building the Website*. cs.odu.edu. Retrieved 26 April 2020, from https://www.cs.odu.edu/~zeil/cowem/Public/buildingTheWebsite/index.html.

Zeil, S. (2020, January 21). *zeil / CoWeM - Course Websites from Markdown*. GitLab. From https://git-community.cs.odu.edu/zeil/Course_Website_Management.

Parenthetical citations - this part and below should not appear in any further work it is for reference only

(J. Brunelle, personal communication, March 2, 2020)

(T. Kennedy, personal communication, February 12, 2020)

Additional requirements for formatting can be found at:

https://www.cs.odu.edu/~tkennedy/cs411/s20/Public/grammarNotes/index.html

Additional requirements for style APA 7