**Curriculum Editor with Automated Visualization**

|  |  |
| --- | --- |
| Alo, Renjiro Vince P.  Alliance of Computer Science Students  Los Banos, Philippines  rpalo1@up.edu.ph | Dunan, Emmanuel  Pasig City, Philippines  ecdunan@up.edu.ph |

# ABSTRACT

With the emergence of K-12 in the Philippines, the academe will have to make a major revision in the curricula of most degree programs. The curriculum creation and revision methods of most universities are still inefficient due to the lack of automation. The study aims to create a web application that automates and visualized the creation of a curriculum that provides the user an easier time and experience for efficiency. The application will be able to detect course conflicts and import and export the curriculum and also allows the students to provide evaluation for specific courses. The application will be evaluated based on the effectiveness of specific features that will be tested randomly by instructors and professors from various colleges with varying years in the academe. A baseline score of 75% will be used to determine if a certain feature in the application is effective or not.

# INTRODUCTION

Organizing, arranging, and preparing various information is a task that usually take a lot of time to complete because of the thorough amount of work done on its preparations and revisions. In most scenarios, organizing information is connected to other and more important tasks since it is part of the preparation of some bigger work. In addition to being vital parts of a system, tasks that are concerned with organizing and planning remain inefficient because of the lack of automation in some of these tasks that can be completed with the help of technology. Preparations have corresponding implementations and most of these implementations are still open to revisions. Most revisions are done to update information and to correct conflicts and problems related to some parts of the system, and some of the conflicts only appear because of lack of consideration in specific conditions allowing certain conflicts to remain undetected during the implementation. This is similar to why class schedules and curricula are very difficult to implement since there are so many factors to consider that will prevent the appearance of conflicts that can mess up the entire system. And with the implementation of K-12 here in the Philippines, universities will have to adjust and revise their current curricula.

A change in curriculum will mean that thorough preparations will be in done once again, and it will mean that a lot of time will be consumed in the process given the fact that most universities are manually writing and editing curricula that are subject to changes. Aside from the manual revisions, course prerequisites also appears to be a problem as these prerequisites can potentially mess up an entire curriculum because of conflicts that can arise in the different courses therefore forcing curriculum revisions to start all over again or undo a number of revisions that were previously made thus consuming more time and effort. Given the idea that people prefer to do things in a much more convenient and efficient way, the evident inefficiency of such preparations suggests an implementation that can provide an easier way of creating and revising curricula.

This inefficiency can be resolved by automating the process of creating a course curriculum. By creating a web application that can generate a visualization of a curriculum with a user-friendly interface, users will have a better experience in creating and revising curricula. Curricula will be generated based on user inputted courses with specified operations to indicate prerequisites and other features. The application will be able to detect conflicts regarding different courses like prerequisites, and revisions will be done through drag-and-drop for an easy experience. Through an application that will be able to visualize and generate curricula, the university will be able to adapt to major changes in education easier, and the computer science community will be more familiar with concepts and approaches regarding visualizing and organizing information.

# RELATED WORK

Several studies have already been conducted prior to the development of a curriculum visualization application. These applications were developed because such visualizations are essential for freshman orientation and for discussion about curriculum revisions by faculties.

In 2007, Sommaruga and Catenazzi used a 3D environment in visualizing curricula. They used XML for the curriculum's data and transformed it to X3D format [8]. By 2008, Getswicki started working in an application called CurricVis (Curriculum Visualization). His approach uses a three-tier software architecture consisting of the 1) knowledge base which is responsible for translating the input curriculum's data into a set of rules, 2) the visualization generation layer which is responsible for generating the directed graphs based from the knowledge base's output and 3) the user interface layer which allows users to select which curriculum they want to view [4]. By 2009, Zucker created a java based application called ViCurriAs (Visual Curriculum Advising System) that is divided into two modules. The Curriculum which allows faculty to create a new curriculum using a drag and drop user interface and the Advising which allows assigning of grades and scheduling information on the specified curriculum [9]. In 2012, Auvinen created an application with the purpose of not only visualizing curriculum data in a more graphical way but also to increase student's motivation in studying. This was done by enabling students construct their own study plans [2]. And in 2014, Aldrich P. implemented The Curriculum Prerequisite Network and used directed graphs to visualize curricula. He used weights to represent dependencies among courses. For example, if passing a certain course A or course B is required to be able to take course C, then both A and B will be given a weight of 1. On the other hand, if passing both of the courses are required to be able to take course C, then the weight will be split among them giving each course a weight of 0.5 [1].

The use of DAG (Directed Acyclic Graph) also appears as a feasible solution to other problems like project resource management, cost accounting, and even in the field of medicine.

In 2003, Bessler, Yang, & Wongcharupan made use of error correction models and DAG to determine dynamic pricing of wheat among its major producing areas. [3] In the same year, Bessler and Haigh also used error correction models and DAG in analysing price discovery questions between commodity markets and transportation market [5]. By 2008, Merrelec, Ciuciu, Isaac, & Benali made use of DAG in their models and showed how it conveniently does the preparatory steps for Gibbs sampling from the model specification [7]. And in 2009, Ishaque, Zaidi and Levis used directed acyclic graphs to manage systems engineering projects. The use of DAG can drastically improve project management because it enables one specification of real-time milestones and breaks the finish-start barrier in activities [6].

The preceding reviews present meaningful information on different curriculum visualization tools implementation, and other problem domains that can be solved using DAG. The goal of the study is to create a web-based curriculum visualization/creation tool with a drag and drop user interface. Curricula can be imported and exported in CSV format and has conflict resolution mechanisms that can detect prerequisite anomaly and if the semester the course is offered matches the one on the curriculum's schedule. The application will also enable PDF generation of curricula for printing purposes. It will also support public evaluation features enabling students and faculties to comment on courses and curriculum designs.

# METHODOLOGY

The development of the project can be divided in to five parts. The conversion of a text input file to a visual representation, the drag and drop implementation for the User Interface (UI), the conflict checking mechanisms, the PDF generation that will allow curricula to be exported in a printable format and lastly, a public evaluation feature.

The data structure to be used in the implementation of the program is a Directed Acyclic Graph. Using DAG fits the problem domain because graphs are normally used to represent networks. Various data can be added and connected to certain nodes to illustrate relationships and constraints, similar to how course curricula are represented. Networks and linking of different variables are used in a course curriculum for varying purposes like what time of the year a certain course is offered and the weighing of some courses to keep track of prerequisite and co-requisite relationships. The said methods will also be adapted in this study to satisfy the project specifications. The effectiveness of using DAG is further supported by recent studies in the field of curriculum visualization because of the same approach used in past projects.

### Import/Export of Files

Files that have comma-separated value (CSV) format will be used to import an existing curriculum that will also be open to revision. Exporting to CSV files will also be enabled for the user. An optimal search algorithm will be used to parse the CSV files for conversion to a graphical visualization of courses that are organized according to the year and semester. The application will also support the generation of Portable Document Format (PDF) to allow the users to obtain the visualized curriculum as it is.

### User Interface

The target user interface to be developed for the project should give the easiest experience for the users. The interface will therefore consist mostly of an empty space for the visualization of the curriculum with a few buttons for file transfer and course manipulation. Courses and prerequisites are represented by shapes and arrows respectively controlled through a drag and drop functionality.

### Conflict Checking

Conflict checking will run in the background while the user is designing a curriculum. An interval will be set to monitor if certain constraints are violated like conflicts between a curriculum map's schedule and the particular semesters a course is offered.

Prerequisite and co-requisite constraints will also be detected in this module. The approach in handling such scenarios is making use of weights for each course [1].

### Public evaluation

Being a web-based application, integration to other web services was kept in mind in planning the implementation of the project. If user accounts will be supported, participants will be recognized using their e-mail address uniquely generated by their own university when commenting on a particular curriculum design or course. This is done to prevent anonymous posts that can ruin the credibility of the public’s opinion.

# EVALUATION

With the implementation of K-12 in most of the schools in the Philippines, the first few years of adaptation will be most crucial as this will determine the succeeding plan of actions of the universities and agencies concerned with education. Changes in course curricula should therefore be effective and efficient since these can greatly affect the whole education of the K-12 graduates of the first few years. Given that this is the objective of the research, an evaluation procedure will be conducted to ensure that the implemented solution is effective.

Keeping in mind that creating or revising a course curriculum is very difficult, the evaluation procedure aims to gather the response of faculty members regarding their satisfaction on the implemented curriculum visualization application as it is and relative to the current manual process of curriculum generation in the form of ratings from a questionnaire that asks the faculties to score the specific features of the said application.

The evaluation will be composed of specific questions pertaining to the different features of the application. The features will be evaluated based on the participant’s opinion regarding the usefulness and effectiveness of the application. The participants will be instructed to rate the specific features from a scale of one to five with five rated as the most effective and one indicating that the feature is not very effective. The total rating per question will be analysed to determine if certain features that are implemented in the application are effective. The evaluation procedure will be tested in the University of the Philippines Los Banos (UPLB). To determine whether a certain feature is effective, the total score must be greater than or equal to the baseline score which is set to 75%. It is necessary for the baseline to be set at a high value since it is the participants—the faculty themselves that will use the implemented solution therefore the solution must be optimized for the users own experience as it is the purpose of the project.

Given that the faculty of a certain university are constantly changing and adapting, the participants of the evaluation procedure will be forty-five instructors and/or professors chosen at random within various colleges in UPLB. Taking into consideration the varying opinions of relatively new instructors and of professors that have witnessed different batches of students, the participants are divided into three groups based on their number of years in the academe. The groups are divided as follows: less than 4 years, 4-7, and 8 or more years in the academe. Each group will therefore have fifteen participants taken from various colleges totalling forty-five participants.

# Timeline

|  |  |
| --- | --- |
| **DATE** | **ACTIVITY** |
| January 25 – February 12 | Drag and drop UI |
| February 15 – February 29 | Import/Export CSV files |
| March 1 – March 11 | Conflict checking functions implementation |
| March 14 – March 24 | PDF Generation and curriculum comments |
| March 25 – April 1 | Debugging and Refactoring |
| April 4 – April 8 | Evaluation and testing |
| April 11 – April 15 | Polishing of project |

# References

[1] Preston R. Aldrich. 2014. *The curriculum prerequisite network: a tool for visualizing and analyzing academic curricula.* Department of Biological Sciences, Benedictine University, USA

[2] Auvinen, T. (2011, November). Curriculum development using graphs of learning outcomes. In *First EUCEET Association Conference New Trends and Challenges in Civil Engineering Education, Patras, Greece* (pp. 27-37).

[3] Bessler, D., Yang, J., & Wongcharupan, M. (2002). Price dynamics in the international wheat market: Modeling with error correction and directed acyclic graphs. *Journal of Regional Science*, *42*(4), 793-825.

[4] Gestwicki, P. (2008, October). Work in progress-curriculum visualization. In*Frontiers in Education Conference, 2008. FIE 2008. 38th Annual* (pp. T3E-13). IEEE.

[5] Haigh, M. S., & Bessler, D. A. (2004). Causality and price discovery: An application of directed acyclic graphs\*. *The Journal of Business*, *77*(4), 1099-1121.

[6] Ishaque, M., Zaidi, A. K., & Levis, A. H. (2009). Project management using point graphs. *Systems Engineering*, *12*(1), 36-54.

[7] Marrelec, G., Ciuciu, P., Pélégrini-Issac, M., & Benali, H. (2003, January). Estimation of the hemodynamic response function in event-related functional MRI: Directed acyclic graphs for a general Bayesian inference framework. In*Information Processing in Medical Imaging* (pp. 635-646). Springer Berlin Heidelberg.

[8] Sommaruga, L., & Catenazzi, N. (2007, April). Curriculum visualization in 3D. In *Proceedings of the twelfth international conference on 3D web technology*(pp. 177-180). ACM.

[9] Zucker, R. (2009). ViCurriAS: a curriculum visualization tool for faculty, advisors, and students. *Journal of Computing Sciences in Colleges*, *25*(2), 138-145.