Computer and Information Science; Vol. 15, No. 1; 2022
ISSN 1913-8989 E-ISSN 1913-8997
Published by Canadian Center of Science and Education

A Decision Support System for Undecided and Exploratory Students

Sobitha Samaranayake[1], Athula D. A. Gunawardena[1], & Robert Meyer[2]

[1] Department of Computer Science, University of Wisconsin-Whitewater, Whitewater, USA

[2] Department of Computer Science, University of Wisconsin-Madison, Madison, USA

Correspondence: Sobitha Samaranayake, Department of Computer Science, University of Wisconsin-Whitewater, Whitewater, WI 53190, USA.

Received: October 14, 2021 Accepted: November 10, 2021 Online Published: November 30, 2021

doi:10.5539/cis.v15n1p1 URL: https://doi.org/10.5539/cis.v15n1p1

Abstract

Choosing a major can be a difficult choice for undeclared students, depending on personal interests, job markets, program costs, and the complexity of graduation requirements. The current procedure for exploring possible majors/minors involves a tedious and time-consuming exploration of static data presented in a convoluted format. This work considers the complexity of degree requirements and presents the design and implementation of an efficient decision support system (DSS) for exploring majors and/or minors. The DSS presents a tool that enables college students to map their already completed courses to degree requirements and then view the majors/minors that yield the shortest path for graduation. Secondly, the DSS system provides data visualization tools to assist students in selecting courses that satisfy the remaining requirements of a selected major/minor.

Keywords: decision support system, undecided major, data visualization, college degree planning

1. Introduction

It is estimated that 20 to 50 percent of students enter college in the U.S. as undecided or undeclared, without having decided on a focus for their studies, and more than 50 percent of students change their major at least once before graduation (Gordan & Steel, 2015). There are many reasons for the indecision of college students, including decision-making difficulties, gender differences, cultural differences, indecisive students, and types of career indecision (Soria & Stebletin, 2013). Many undecided students are skeptical about how their personal strengths and limitations relate to coursework required in particular majors. Furthermore, choosing a major may depend on personal interests, job market, program cost, or the complexity of graduation requirements (Pozzebon, Ashton, & Visser, 2014). Many students make initial choices based on their interests but change their majors because of changing career interests or academic interests (Bullock-yowell, McConnell, & Schedin, 2014).

The key to graduating in four years is choosing a major early and sticking with it. If undeclared students take too long to declare a major, then it may affect their intended graduation date. In addition, changing majors may keep students in college past their intended graduation date and drive up their debt. Hence, higher education institutions have prioritized supporting undecided students with their major and career decisions. Wang and Orr (2019) used data analytics to inform decision-making in academic advising and support undecided students' academic success. Halasz and Bloom (2019) examined the resources students identified as most valuable and the factors most influential in their decision to transition out of selective majors. Streufert (2019) investigated the effects of alternative advising, such as coping with loss, managing anxiety, and restoring self-efficacy, on renewing focus of undeclared students so that they stay focused and graduate on time. Marade and Brinthaupt (2018) examined reasons for students to change a college major. Iyer and Variawa (2019) used supervised Machine Learning classification algorithms to analyze the potential inclination of the undecided/undeclared first-year engineering students at the University of Toronto. Glaessgen, MacGregor, Cornelius-White, Hornberger, and Baumann (2018) examined the challenges and experiences of first-generation undecided students transitioning to a new and unfamiliar academic environment. The relationship between academic major change and ten personality traits (the five broad and five narrow traits) was investigated in Foster (2017). Ongoing research explores different strategies that support undecided students in their major and career decisions. However, there is a dearth of degree planning for comparing different major/minor combinations.

Undeclared students may wish to find, depending on their completed and transferred courses, possible major(s) that better align with their interests and career opportunities. From an algorithmic perspective, choosing such a major is a difficult combinatorial optimization problem due to the complexity of degree requirements, prerequisite dependencies, and course rotation policies. Developing a mechanism to generate a set of feasible majors, depending on a set of completed courses and desired graduation date, is thus a complex problem that involves comparing a large number of possible degree paths.

Academic advisors are either faculty or professional advisors, and they use academic planning tools to help students plan their degree completion and make decisions about many other academic related activities. Academic Advising Report (AAR) or Degree Progress Report (DPR) is the official document used for advising and the information related to degree requirements, course schedules, and course prerequisites are stored in static documents or PDF files. Hence, it is a tedious and time-consuming process to compare requirements for different majors and a formidable task for undeclared students to choose a major so that they could graduate in a timely manner. The new generation of students are digital natives who expect advising resources to be online and available in a user-friendly format but it is a challenge to develop tools that can analyze student’s progress toward completing degree requirements due to the complexity of degree requirements and prerequisite dependencies.

Degree requirements are tied to completing a set of courses or units satisfying the requirements. Some of the degree requirements may refer to additional constraints such as course level (lower-division vs. upper-division) or student’s minimum grade point average (GPA). Additionally, some of the courses may not be available until a minimum number of units has been earned. Some of the academic institutions may count courses only once in the major or minor, either as a required course or as an elective, but not as both. There may be hidden prerequisites (i.e. prerequisites of a prerequisite course that may not be explicitly listed as a part of any other degree requirements). Furthermore, there may be additional conditions such as choosing major/minor emphasis areas. Degree requirements are often defined in terms of a set of course requirements that covers particular subjects or areas of expertise. Many courses list prerequisites that are defined using a list of courses, all of which or a subset of which must be completed successfully in order to satisfy the prerequisites. Furthermore, a few of the prerequisites may include constraints on course grades to ensure students gain the necessary proficiency for receiving the maximum gain from the next course.

Every existing academic planning tool is capable of evaluating and displaying only a single major/minor combination or double major combination. Since it is difficult to define a standard format for representing degree requirements, most of the existing academic planning tools use custom-made systems that are difficult to scale up. The current procedure for exploring possible majors/minors involves a tedious and time-consuming exploration of static data presented in a convoluted format. Degree requirements vary from department to department and from college to college, and information pertaining to requirements for different majors/minors is often scattered in various places.

2. Related Work

There has been an interest in designing interactive and visualization tools for academic advising. Marques, Ding, and Hsu (2001) presented a design and development of a web based academic advising system. Gutiérrez et al (2018) presented a design and implementation of a Learning Analytics Dashboard for Advisers, LADA, to support the decision-making process of academic advisers through comparative and predictive analysis. Moreno, Bischof, and Hoover (2012) presented an interactive visualization tool for exploring course dependencies between courses. Dechter (2007, 2009) introduced an integer linear programing model for finding academic plans that would satisfy a given set of graduation requirements and other constraints in the shortest possible time. Kowalski and Ealy (1991) used artificial intelligence to design an expert system for the advisement of two-year community college students. Horvth, Molontay, and Szab (2018) presented an efficient visualization tool to analyze student flow patterns to track retention and graduation rates. Kumar (2017) presented a prototype of a spreadsheet-based scheduling model to create individual graduation roadmaps.

Prerequisite visualization tools are extremely useful for preparing academic plans. Aldrich (2014) used the overall topology of the courses at Benedictine University to propose a directed acyclic graph for representing prerequisite relations where each edge represents a logical relationship such as all of or one of. Zucker (2009) presented a curriculum visualization tool for developing and arranging the flow of courses for a particular program. Chen and Siyuan (2017) presented an interactive course selection scheme with prerequisite hierarchy. Their work includes visualization of all of, one of, or either or logical relationships of courses offered at University of British Columbia. Samaranayake and Gunawardena (2020) introduced a graphical data visualization tool that enables students and advisors to easily understand course prerequisite structure and to readily determine paths that lead to the satisfaction of degree requirements.

There has been an interest in designing DSS for course planning. Siddiqui, Raza, and Tariq (2018) introduced a web-based group DSS for academic term preparation at a business college of a large Middle Eastern university. Roushan et al. (2013) presented a DSS for course planning. Miranda, Rey, and Robles (2012) developed a web-based DSS for course and classroom scheduling. Oladokun and Oyewole (2015) presented a DSS for university admission seekers. Al-Qaheri, Hasan, and Al-Husain (2011) presented a DSS for a course scheduler. Most of these decision support systems are designed for course scheduling. Meyer, Gunawardena, Samaranayake, Deshpande, and Premadasa (2021) presented an application of information technology in providing decision support in college planning.

The complexity of degree requirements, prerequisite dependencies, and user preferences make the automated degree planning problem an inherently hard combinatorial optimization problem. Due to its complexity, the present commercial degree planning systems have avoided automation and limited their features to semester by semester drag and drop course selections. Integer programming models to generate degree plans with simplified requirements have been proposed in Dechter (2007, 2009). Although these models are useful for calculating lower bounds for comparison, they are intractable for practical systems which deal with complex degree planning problems with various constraints and are expected to provide fast solutions.

This paper will focus on identifying a data structure and a mechanism capable of evaluating degree requirements and then developing a DSS for exploring majors/minors. First, we identify possible degree requirement types and define a data structure that can represent such requirements. Then we create a DSS that helps students select a major along with the courses needed to satisfy additional degree requirements beyond those for the major. These additional requirements often include the selection and completion of a minor in a subarea and the completion of so-called general education (GE) requirements spread across a wide variety of topics.

To demonstrate the potential of the decision support system, we describe an implementation based on degree requirements and majors/minors offered at University of Wisconsin-Whitewater (UWW), a mid-sized U.S. public institution with about 11,000 students, which has 574 pages of course information and degree requirements covering more than 150 possible majors and about 120 possible minors. A student generally has a choice of focus areas within the major hence there are about thousands of different course sequences that meet requirements of a selected major at UWW. In addition to deciding upon and meeting the requirements of both a major and minor, a student wishing to complete a degree is faced with the challenge of selecting courses that meet general education requirements that specify the minimum number of credits needed in a variety of additional subareas, such as Communication Skills, Calculation Skills, Quantitative and Technical Reasoning, Cultural heritages, World of Ideas, Communities, Physical Health and Well-Being, and Racial/Ethnic Diversity.

3. Data Structure for Processing Degree Requirements

Many of the college degree requirements are expressed in terms of number of courses, units, or credits that must be completed to fulfill each requirement. College degree requirements are often expressed using one or more of the following conditions:

- Complete a predefined list of courses.
- Complete a subset from a set of elective courses.
- Complete a specified number of courses from each of several lists of courses.
- Select a subset of lists and then complete a specified number of courses from each of the selected lists (e.g., select three of five course lists and then complete one course from each selected list).
- Complete a specified number of courses from a selected subset of lists (e.g., complete a total of three courses from at least two different sets).
- Complete courses with a specified total number of units from a list of courses.
- Complete a specified number of units from a selected subset of lists (e.g. complete at least six units from two different lists).

First, we define an appropriate data structure for evaluating degree requirements. A typical degree requirement belongs to one of the following categories:

- Type A: complete k courses from a specified set of courses
- Type B: from a specified set of courses, complete at least m courses/units, but no more than n courses/units will be counted toward major/minor requirements
- Type C: complete at least k units from a specified set of courses

- Type D: combination of Type A, Type B, and/or Type C requirements

Type A, Type B, and Type C degree requirements are relatively easy to represent but Type D requirements are often complex and difficult to represent. There may also be other requirements, such as GPA requirements, minimum number of credits/units needed to complete, internships, capstone projects, etc. Samaranayake and Gunawardena (2020) introduced a generic requirement structure, named basic requirement, that is able to represent most of the college degree requirements. In this work, we extend the definition of the basic requirement to represent almost any degree requirement found in a college catalog:

Definition: A **basic requirement** is a 7-tuple (A, T, C_{min}, C_{max}, U_{min}, U_{max}, δ), where

(1) A is a set of objects,

(2) T is the type of requirement (Type 1: select number of objects; Type 2: select number of units; Type 3: combination of Type 1 and Type 2, etc.),

(3) C_{min} is the lower bound of objects,

(4) C_{max} is the upper bound of the objects,

(5) U_{min} is the lower bound of the units,

(6) U_{max} is the upper bound of the units (note that this upper bound is simply a limit on the number of units counted toward a particular requirement rather than a limit on the total number of units that may be taken with respect to a set of courses), and

(7) $\delta: A \rightarrow \{1, 0\}$ is a function such that $\delta(A) = 1$ if A is a major/minor unit -counting set of objects and $\delta(A) = 0$ otherwise. (This function is used to prevent duplicate counting of units.)

There may be other requirements that specify the minimum number of courses/units needed for a degree or in a subarea or for higher-level courses. These are called special requirements.

Type A, Type B, and Type C requirements are basic requirements. Type D requirements can be represented using a set of basic requirements. Hence, requirements for any major/minor M_i are expressed as $M_i = \{R_{i1}, R_{i2}, \dots, R_{ir}\}$ where each degree requirement $R_{ij}(A, T, C_{min}, C_{max}, U_{min}, U_{max}, \delta)$ is either

- a basic requirement where A is a set of courses or
- a basic requirement where A is a set of basic requirements.

Let R_{ij}(A, T, C_{min}, C_{max}, U_{min}, U_{max}, δ) be a basic requirement. An object a_i (course or a basic requirement) satisfies a basic requirement R_{ij} if $a_i \in A$. We define a boolean function on A, $b_i: A \rightarrow \{1, 0\}$ such that $b_i(a)$ = 1 if $a \in A$ and $b_i(a) = 0$ if $a \notin A$.

A set $A = \{a_1, a_2, \dots, a_k\}$ satisfies a basic requirement R_{ij} if R_{ij} is of Type 1 and $C_{min} \leq \sum_{j=1}^{k} b_i(a_j) \leq C_{max}$ or R_{ij} is of Type 2 and $U_{min} \leq \sum_{j=1}^{k} b_i(a_j) * u(a_j)$, where $u(a_j)$ is the number of units of a_j.

Any set of degree requirements can be expressed using a set of basic requirements. Consider a set of requirements for a major/minor M_i defined as follows:

R_{i1} : complete one of the courses $C1$ or $C2$

R_{i2} : complete all of the courses $C3, C4, C5,$ and $C6$

R_{i3} : complete 6 units from the courses $C7, C8, C9, C10$

R_{i4} : Complete 6 – 12 units with at least two units in $A_1 = \{C11, C12, C13\}$, at least three units in $A_2 = \{C14, C15, C16\}$, and one unit in $A_3 = \{C17, C18\}$

The requirements $R_{i1}, R_{i2},$ and R_{i3} are basic requirements and the requirement R_{i4} may be expressed using a combination of basic requirements as

$R_{i4} = (\{R_{i41}, R_{i42}, R_{i43}, R_{i44}\}, 3, 4{,}4{,}6{,}12{,}1)$ where$R_{i41} = (\{C11, C12, C13\}, 2, 0, 0, 2, 2, 0)$,

$R_{i42} = (\{C14, C15, C16\}, 2, 0, 0, 3, 3, 0)$
$R_{i43} = (\{C17, C18\}, 2, 0, 0, 1, 1, 0)$, and

$R_{i44} = (\{C11, C12, \dots, C18\}$, *2, 0, 0, 6, 12, 0*).

4. Exploring Major and Minor Paths

When choosing a major or a minor, a student would normally have completed some courses that may count toward satisfying requirements for some of the majors or minors. Now we define an efficient process for mapping completed courses to prospective majors or minors.

Let S be the set of all courses, M be the set of all majors, and N be the set of all minors offered by a degree-granting institution. A major path is a minimal set of courses that satisfies all the requirements of a college major and a minor path is a minimal set of courses that satisfies all the requirements of a college minor. Let $M_i \epsilon M$ and $M_i^p = \{P_{i1}, P, \dots, P_{iq}\}$ be the set of all possible major paths for the major M_i. It is possible to generate all possible major paths for the set of all the majors, M, offered at a given institution. The process begins with identifying a matching set of possible courses $C(M_i^R)$ that can play a role in completing the degree requirements M_i^R and then exploring all possible major paths for each major $M_i \in M$. There are about 15,000 different course sequences that meet the Computer Science major requirements for a BS degree at UWW. Hence, there could be millions of possible major paths for the entire collection of majors so it would be a daunting task to create and maintain such a collection of possible major paths. Thus, we first describe a process for identifying the subset M^C of M that corresponds to the set of majors for which at least one requirement has been partially satisfied

Let $V = \{(s, m) \mid s \in S, m \in M\}$ be the set of all ordered pairs that connect courses to majors in the sense that s appears in at least one major path of M. In many universities, a single department offers majority of the courses satisfying a given major. Some courses offered by a department are not part of any major while courses from a particular department may satisfy requirements of majors outside the department. Therefore, V is a relatively small subset of $S \times M$.

Let C be a set of courses completed (taken/waived) by a particular student and

$V_c = \{(s, m) \mid s \in C, m \in M\}$ be the subset of V that defines a mapping between the set of completed courses C and the set of majors M.

Let $M_i^R = \{R_{i1}, R_{i2}, \dots, R_{ir}\}$ be the set of requirements for the major $M_i \in M$, where each requirement in R_{ij} is a predicate defined on a subset of S. Courses in the set C may partially satisfy a subset of the requirements associated with the subset M^c of M and a subset N^c of N so that M^c consists of majors accepting some or all courses in C and N^c consists of minors accepting some or all courses in C.

Let $C = \{c_1, c_2, \dots, c_n\}$ denote the set of courses already completed by a specific student. Using the set V_c , we can find an ordered list of majors $M^{c_i} = \{M_{i0}, M_{i1}, M_{i2}, \dots, M_{ij}\}$ where the course c_i partially satisfies one or more requirements of each M_{ik} in M^{c_i} . Then $M^C = \bigcup_{i=1}^{n} M^{c_i}$. Thus, we can easily identify the set M^C for a given set of completed courses C.

We now discuss the process for using completed courses to evaluate remaining requirements for majors in M^C.

Suppose $M_i = \{R_{i1}, R_{i2}, \dots, R_{ir}\} \in M^C$. Then the number of courses involved in the requirement $R_{ij}(A, T, C_{min}, C_{max}, U_{min}, U_{max}, \delta)$ is $\sum_{k=1}^{n} b_{ij}(c_k)$; the total number of units counted toward a requirement R_{ij} is $S_j = \sum_{k=1}^{n} b_{ij}(c_k) * u(c_k) * \delta$ where $u(c_k)$ is the number of units of the course c_k; and the total number of units counted towards completing the major M_i is $S_{M_j} = \sum_{j=1}^{r} S_j$ where r is the total number of requirements of the major M_i. This calculation can be repeated for each major $M_i \in M^C$ to assess all possible major choices for a given set of completed courses.

Table 1. *Sample course mapping consisting of a set of six courses,* $C = \{c_1, c_2, \dots, c_6\}$

	Major							
Course	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
C_1		R_{21}			R_{52}			
C_2								
C_3				R_{42}, R_{45}			R_{72}	
C_4				R_{42}				
C_5							R_{72}	
C_6				R_{42}			R_{73}	

Table 1 defines a sample mapping between a set of six courses and a set of eight majors. Based on the sample course mapping in Table 1,

$V_c = \{(c_1,\ M_2), (c_1,\ M_5), (c_3,\ M_4), \dots, (c_6,\ M_7)\}$

$M^C = \bigcup_{i=1}^{n} M^{c_i} = \{M_2,\ M_4,\ M_5,\ M_7\}$

Hence, we need to explore major paths of only the four majors in the set M^C, instead of the major paths of set of all majors M.

We use the same process for exploring possible major/minor combinations for each major $M_i \in M^C$. Some universities do not allow courses counted for a major to be used for satisfying minor requirements. Suppose C_i is the set of courses which satisfy some of the requirements for the major $M_i \in M^C$. Then we can use the set $C \setminus C_i$ to explore possible major/minor combinations.

There is usually a minimum unit requirement for graduation. We can use our calculations to calculate the minimum number of additional units needed for graduation with every possible major/minor combination:

Units required = Total units required – (Major units + Minor units + General education units)

Upon exploring possible major/minor combinations, we can generate a set of courses to complete the remaining degree requirements while minimizing the total number of additional units needed for each major/minor combination. Such a process allows us to produce the shortest path for graduation for an undeclared student.

The set V of all ordered pairs that connect courses to majors needs to be computed only once and then needs to be updated with each curriculum change. Usually, curriculum changes occur only once or twice a year. Hence, it is a standard process to update the degree requirements and the set V with every curriculum change. When we extract completed courses for a particular student, we use an algorithm to map completed courses to degree requirements to compute the set $V^c$and then the set M^C. The set M^C consists of all the possible majors that would allow a student to complete the degree requirements in a timely manner. It is possible to find the major or majors that require the least number of units to complete.

5. Data Visualization Method

Major requirements, prerequisite conditions, and course rotations must be taken into consideration when planning courses for the completion of the degree. In general, prerequisites are completed/waived/transferred courses or test scores that must be completed before taking a specific course, and some of the prerequisites are tied to course grades and courses from other disciplines. Table 2 displays an example of the prerequisite conditions for a sample set of seven courses.

Table 2. Prerequisite conditions for a sample set of courses

Courses	**Prerequisites**
$C2, C3, C4$	$C1$ with a grade of C or better
$C5, C6$	$C2$ with a grade of C or better or $C4$ with a grade of B or better
$C7$	$C2$ with a grade of C or better or $C3$
$C8$	$C7$ or (C5 and $C6$), with a grade of C or better

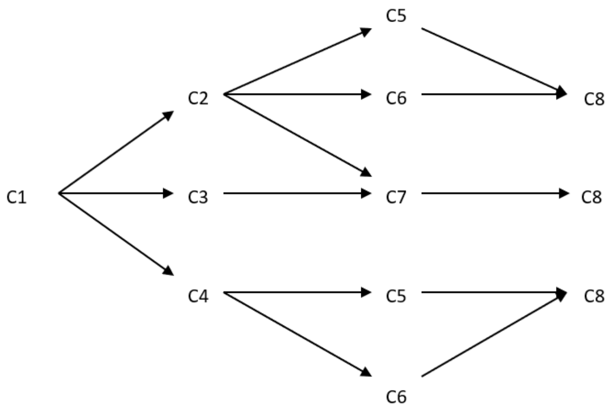


Figure 1. Prerequisite structure

Suppose a student wishes to select a major path that includes the course $C8$. In order to check if course prerequisites for $C8$ are satisfied or to find the shortest path for satisfying the prerequisites, it would be extremely helpful if the prerequisite structure can be visualized using a directed graph.

We use an adjacency matrix of a directed graph $D(V, E)$ to represent course prerequisite structure (CPS) where nodes (V) represent courses and edges (E) represent prerequisite relationships. Table 2 contains information needed to define an adjacency matrix of the directed graphs for the sample set of courses. Figure 1 shows a directed graph depicting the prerequisite structure described in Table 2.

Existing degree planning tools show CPS using static data structures similar to Table 2. It is extremely helpful if the course structure is displayed as a directed graph that can be dynamically updated using completed courses. We utilize novel visualization tools introduced in Samaranayake and Gunawardena (2020) to visualize CPS for degree paths.

6. Implementation

Degree requirements are often specified in terms of course offerings. Therefore, each requirement is stored in a database using an appropriate format suitable for our algorithms. In order to speed up the process, we have designed a suitable data structure for storing each type of basic requirement. We use a relational database to store degree requirements and course information. A set of completed/waived courses is needed to explore major/minor combinations. Figure 2 shows the architecture diagram of the DSS system.

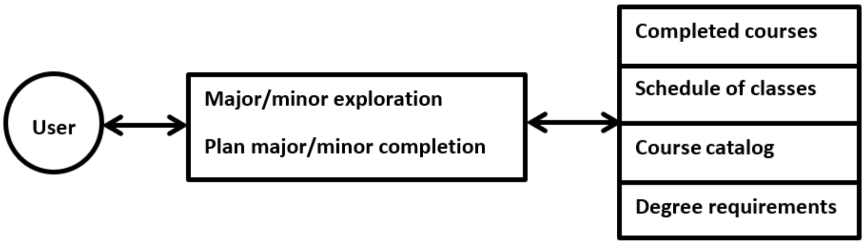
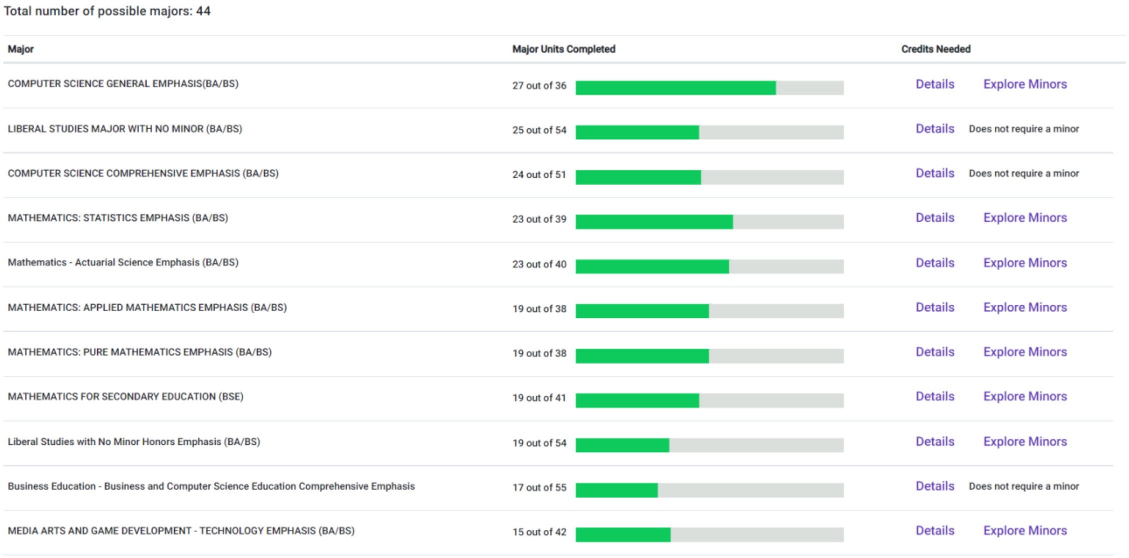


Figure 2. Architecture diagram of the DSS system


Total number of possible majors: 44

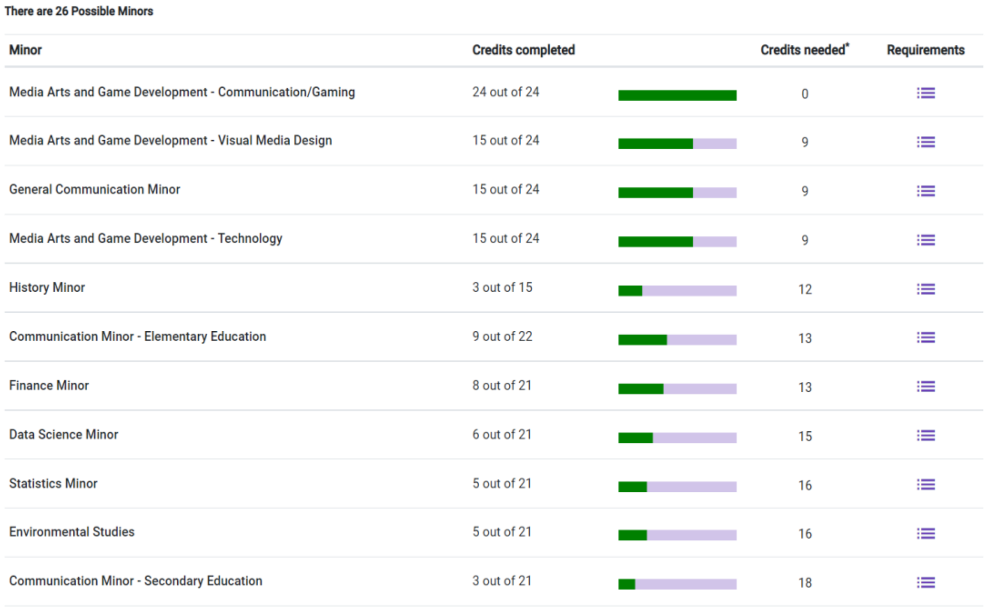
Major	Major Units Completed	Credits Needed	
COMPUTER SCIENCE GENERAL EMPHASIS(BA/BS)	27 out of 36	Details	Explore Minors
LIBERAL STUDIES MAJOR WITH NO MINOR (BA/BS)	25 out of 54	Details	Does not require a minor
COMPUTER SCIENCE COMPREHENSIVE EMPHASIS (BA/BS)	24 out of 51	Details	Does not require a minor
MATHEMATICS: STATISTICS EMPHASIS (BA/BS)	23 out of 39	Details	Explore Minors
Mathematics - Actuarial Science Emphasis (BA/BS)	23 out of 40	Details	Explore Minors
MATHEMATICS: APPLIED MATHEMATICS EMPHASIS (BA/BS)	19 out of 38	Details	Explore Minors
MATHEMATICS: PURE MATHEMATICS EMPHASIS (BA/BS)	19 out of 38	Details	Explore Minors
MATHEMATICS FOR SECONDARY EDUCATION (BSE)	19 out of 41	Details	Explore Minors
Liberal Studies with No Minor Honors Emphasis (BA/BS)	19 out of 54	Details	Explore Minors
Business Education - Business and Computer Science Education Comprehensive Emphasis	17 out of 55	Details	Does not require a minor
MEDIA ARTS AND GAME DEVELOPMENT - TECHNOLOGY EMPHASIS (BA/BS)	15 out of 42	Details	Explore Minors

Figure 3. Initial portion of sample output of possible majors

The current implementation of the DSS is based on degree requirements and majors/minors offered at University of Wisconsin-Whitewater. First, completed courses are mapped to majors to identify the set M^C. Then, requirements for each major $M_i \in M^C$ are evaluated using the completed courses to produce the report in Figure 3.

Figure 3 shows the assessment of a set of 38 courses completed at UW-Whitewater. The UW-Whitewater database consists of 154 majors, 121 minors, and 3300 courses. Students must complete at least 120 units. Some of the majors require an approved minor. Courses counted toward the major cannot be counted for a minor. Students must satisfy general education requirements. In addition, some majors require students to complete a separate mathematics requirement. The output includes a set of possible majors and the number of possible majors. For each such major, it also includes a list of courses satisfying its requirements, courses satisfying the mathematics requirement, and the number of credits needed to complete the major.

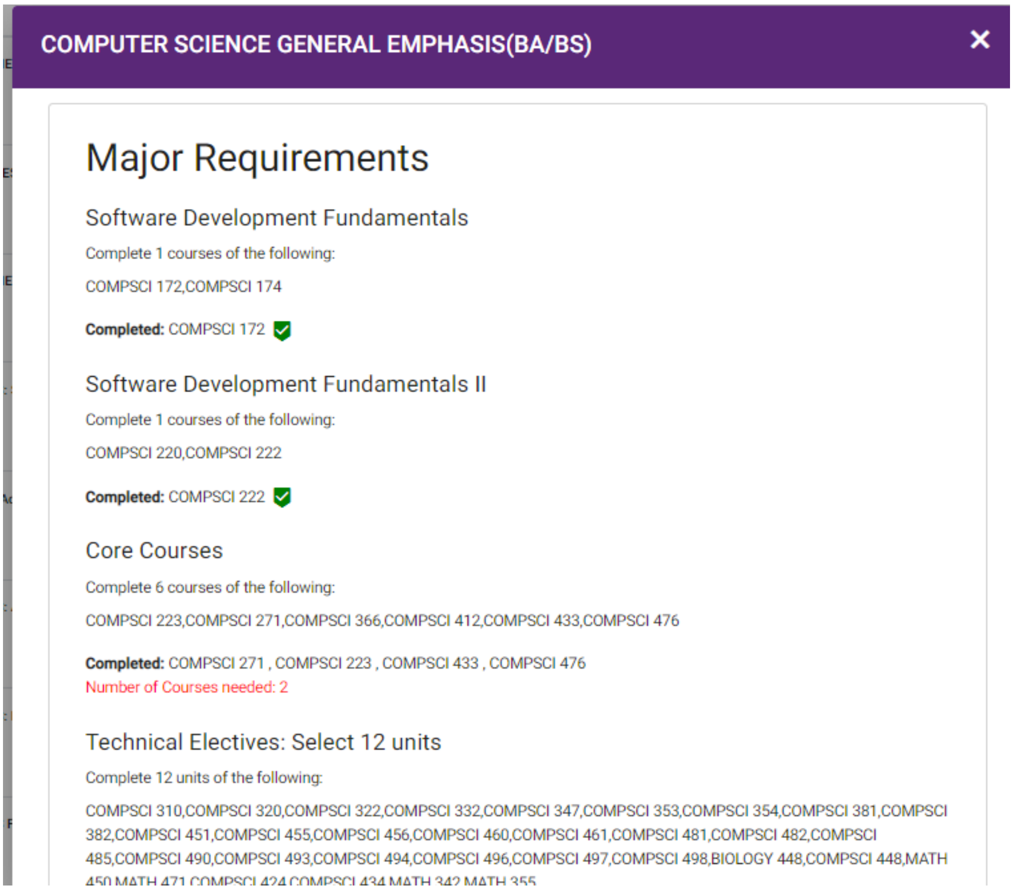
In the sample execution, there are 44 possible majors in the set M^C. The DSS also provides a mechanism to explore minors for those majors that require an approved minor. Figure 4 includes a list of possible minors if the student choses computer science general emphasis (BS) as the major. There are 26 possible major/minor combinations for the selected major.



There are 26 Possible Minors

Minor	Credits completed	Credits needed*	Requirements
Media Arts and Game Development - Communication/Gaming	24 out of 24	0	☰
Media Arts and Game Development - Visual Media Design	15 out of 24	9	☰
General Communication Minor	15 out of 24	9	☰
Media Arts and Game Development - Technology	15 out of 24	9	☰
History Minor	3 out of 15	12	☰
Communication Minor - Elementary Education	9 out of 22	13	☰
Finance Minor	8 out of 21	13	☰
Data Science Minor	6 out of 21	15	☰
Statistics Minor	5 out of 21	16	☰
Environmental Studies	5 out of 21	16	☰
Communication Minor - Secondary Education	3 out of 21	18	☰

Figure 4. Initial portion of the list of possible minors for the selected computer science major



COMPUTER SCIENCE GENERAL EMPHASIS(BA/BS)

Major Requirements

Software Development Fundamentals

Complete 1 courses of the following:

COMPSCI 172,COMPSCI 174

Completed: COMPSCI 172 ✓

Software Development Fundamentals II

Complete 1 courses of the following:

COMPSCI 220,COMPSCI 222

Completed: COMPSCI 222 ✓

Core Courses

Complete 6 courses of the following:

COMPSCI 223,COMPSCI 271,COMPSCI 366,COMPSCI 412,COMPSCI 433,COMPSCI 476

Completed: COMPSCI 271 , COMPSCI 223 , COMPSCI 433 , COMPSCI 476
Number of Courses needed: 2

Technical Electives: Select 12 units

Complete 12 units of the following:

COMPSCI 310,COMPSCI 320,COMPSCI 322,COMPSCI 332,COMPSCI 347,COMPSCI 353,COMPSCI 354,COMPSCI 381,COMPSCI 382,COMPSCI 451,COMPSCI 455,COMPSCI 456,COMPSCI 460,COMPSCI 461,COMPSCI 481,COMPSCI 482,COMPSCI 485,COMPSCI 490,COMPSCI 493,COMPSCI 494,COMPSCI 496,COMPSCI 497,COMPSCI 498,BIOLOGY 448,COMPSCI 448,MATH 450,MATH 471,COMPSCI 424,COMPSCI 434,MATH 342,MATH 355

Figure 5. Major requirement details

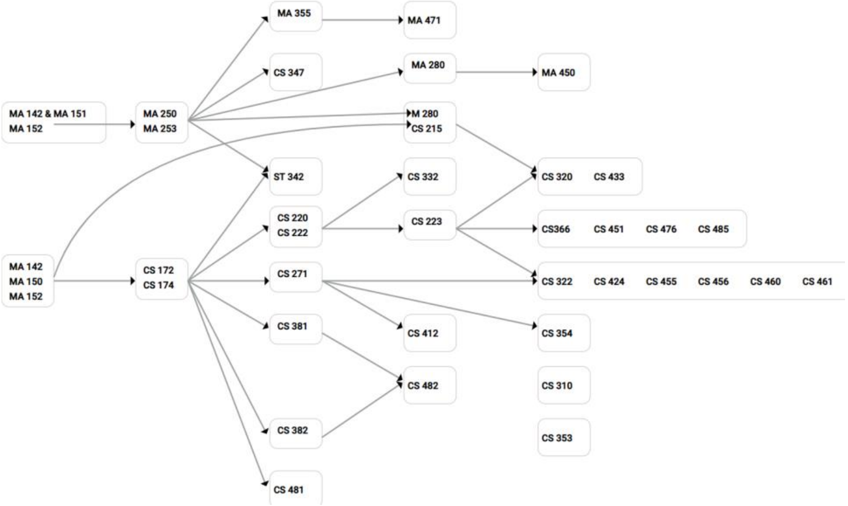


Figure 6. Course prerequisite structure

Furthermore, the DSS provides a mechanism for displaying progress with respect to the requirements for each possible major and minor. Figure 5 includes a list of requirements for the computer science major, general emphasis (BS). A green badge identifies requirements that are fulfilled. If a requirement is not yet fulfilled, then an appropriate message is displayed indicating what is needed to fulfill the requirement.

The sample execution is based on a set of 38 completed courses at University of Wisconsin-Whitewater. Our efficient algorithm was able to map the completed courses to requirements of 44 out of 154 majors in the database and produce a report of those 44 possible majors in less than a second. Although the DSS provides a report that includes the number of credits needed to complete each of the possible majors, students must consider the prerequisite conditions, and course rotations, and other hidden requirements before selecting a major. The DSS system uses an adjacency matrix of a directed graph $D(V,E)$ to store course prerequisite structure (CPS) of each major in a database to display the CPS as a dynamic flow chart that helps students select courses to satisfy the remaining requirements.

Figure 6 shows the CPS for the computer science major at UWW, prior to completing any of the courses in the major. Nodes with a stack of courses represent prerequisite courses where only one of the courses is needed to be taken to satisfy the prerequisite. If two or more arrows are pointing to the same child node, then each of the prerequisite relationships must be satisfied for the course list attached to the child node to be available. CPS is extremely useful for identifying any bottleneck conditions that may prolong the graduation date. For example, CS 223 and CS 271 are perquisite courses for many of the 300-level or higher computer science courses. Hence, their prerequisites must be completed as soon as possible to minimize the time to complete the degree.

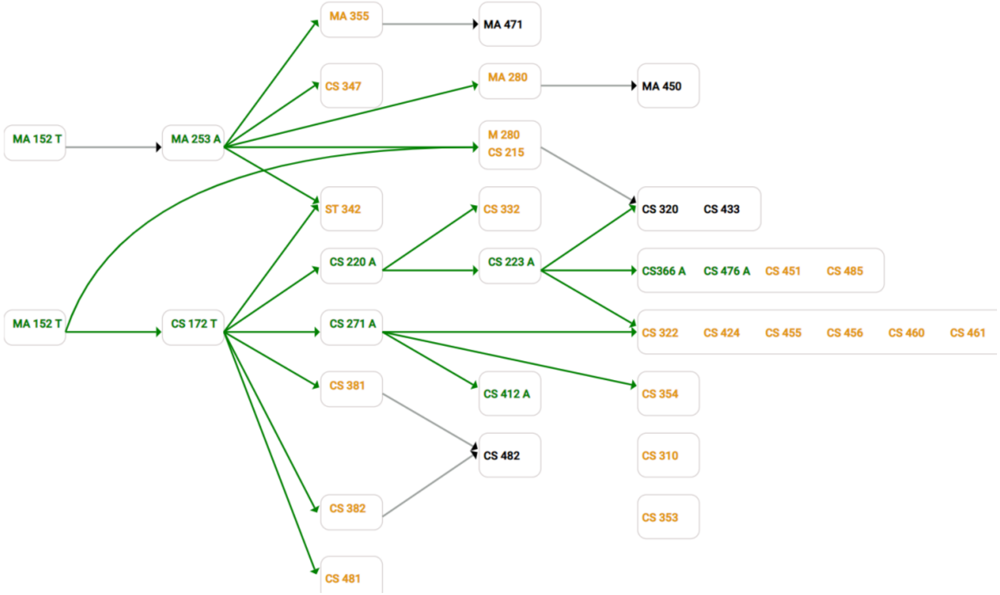


Figure 7. Updated CPS

Figure 7 shows an updated CPS that includes an updated course prerequisite structure, after completing some of the courses required for the major.

The course planning page consists of a course prerequisite structure, list of the degree requirements, an indication of whether each requirement has been satisfied, and courses credited towards satisfying each requirement. The course structure is updated dynamically to narrow down the major path choices, based on the completed courses. The courses shown in green are the courses already completed and the courses shown in orange are the courses whose prerequisites are satisfied. Green arrows point to courses that are available to take, based on the completed courses. In this sample execution, student needs to choose 4 of the 300-level or above courses appear in either orange of black. The CPS is very useful for students to explore the course structure and then select courses to satisfy requirements.

7. Conclusion

This work introduces an appropriate data structure for representing degree requirements and presents the design and implementation of a DSS that helps students explore possible major/minor combinations and find the shortest path for major completion.

The current implementation is based on the UW-Whitewater course catalog of 154 majors, 121 minors, and 3300 courses. Since requirements vary from one major/minor to another, a typical degree mapping application uses separate files for processing individual major/minor requirements. As such, it is a daunting task to create and maintain such an application and there are no such applications for mapping courses to multiple majors/minors. The data structure we introduced in this work is capable of using a single application for processing every major/minor requirement. Hence, this application eliminates the painstaking what-if analysis of static data for exploring possible majors/minors. Furthermore, this application helps undeclared students quickly compare major/minor combinations and find the shortest path for major completion. Although the implementation is based on the UW-Whitewater course catalog, the system can easily be extended to course catalogs at other universities.

The method we introduce in this work for mapping completed courses to majors is a very efficient technique of reducing the complexity of the problem. Hence, such a technique can be very useful in other similar applications. There is a dearth of data visualization tools for displaying college degree-planning information, mainly due to complexity of degree requirements. The new data structure introduced in this research is extremely useful for creating data visualization tools. We have already developed a dependency evaluation and visualization tool using a version of the aforementioned data structure. We plan to expand the current DSS to create an interactive degree personalization system.

References

Aldrich, P. R. (2014). The curriculum prerequisite network: a tool for visualizing and analyzing academic curricula. *arXiv* preprint arXiv:1408.5340.

Al-Qaheri, H., Hasan, M. K., & Al-Husain, R. (2011). A Decision Support System for a Three-Stage University Course Scheduler with an Application to College of Business Administration, Kuwait University. *International Journal of Data Analysis and Information Systems*, *3*, 95-110.

Bullock-yowell, E., McConnell, A. E., & Schedin, E. A. (2014). Decided and Undecided Students: Career Self-efficacy, Negative Thinking, and Decision-Making Difficulties. *NACADA Journal*, *34*(1), 22-34.

Chen, J., & Siyuan, H. (2017). *UBCourse Vis.*

Dechter, A. (2007). Model based student academic planning. *International Journal of Applied Management and Technology*, *5*(1), 87-104.

Dechter, A. (2009). Facilitating timely completion of a college degree with optimization technology. *Association for the Advancement of computing in Educational Journal*, *17*(3), 215-229.

Foster, N. A. (2017). Change of Academic Major: The Influence of Broad and Narrow Personality Traits. *College Student Journal, 51*(3), 363-379.

Glaessgen, T. A., MacGregor, C., Cornelius-White, J. H., Hornberger, R. S., & Baumann, D. M. (2018). First-Generation Students With Undecided Majors: A Qualitative Study of University Reacculturation. *NACADA Journal, 38,* 22-35.

Gordan, V. A., & Steele, G. E. (2015). *The undecided college student: An academic and career advising challenge*. 4th. ed., Springfield, IL: Charles C. Thomas.

Gutiérez, F., Seipp, K., Ochoa, X., Chiluiza, K., De Laet, T., & Verbert, K. (2018). LADA: A learning analytics dashboard for academic advising. *Computers in Human Behavior*, *107*. https://doi.org/10.1016/j.chb.2018.12.004

Halasz, H. M., & Bloom, J. L. (2019). Major Adjustment: Undergraduates' Transition Experiences When Leaving Selective Degree Programs. *NACADA Journal*, *39*(1), 77-88.

Horvth, D. M., Molontay, R., & Szab, M. (2018). Visualizing student flows to track retention and graduation rates, 2018 *22nd International Conference Information Visualisation, IV,* 338-343.

Iyer, S., & Variawa, C. (2019). Using machine learning as a tool to help guide undeclared/undecided first-year engineering students towards a discipline. *Proceedings of the Canadian Engineering Education Association* (CEFA-ACEG19) Conference. https://doi.org/10.24908/pceea.vi0.13741

Kowalski, K., & Ealy, D. (1991). Schedule advisement expert system. *Computers in Human Behavior*, *17*(4), 259-265.

Kumar, R. (2017). A Spreadsheet-based Scheduling Model to Create Individual Graduation Roadmaps. *Journal of Supply Chain and Operations Management*, *15*(2), 165-188.

Marade, A. A., & Brinthaupt, T. M. (2018). Good and bad reasons for changing a college major: A comparison of student and faculty views. *Education*, *138*(4), 323-336.

Marques, O., Ding, X., & Hsu, S. (2001). Design and development of a Web-based academic advising system. Proceedings of the 31st Annual Frontiers in Education Conference. *Impact on Engineering and Science Education*. https://doi.org/10.1109/FIE.2001.964053

Miranda, J., Ray, P. A., & Robles, J. M. (2012). udpSkeduler: A Web architecture based decision support system for course and classroom scheduling. *Decision Support Systems*, *52*(2), 505-513.

Moreno, G. A., Bischof, W. F., & Hoover, H. J. (2012). Interactive visualization of dependencies. *Computers and Education*, *58*, 1296-1307.

Meyer, R. M., Gunawardena, A. D. A., Samaranayake, S., Deshpande, V., & Premadasa, K. (2021). Information Technology for Student Decision Support in College Planning. *Computing Technologies and Applications: Paving Path Towards Society 5.0*. Boca Raton, FL, USA: CRC, 243-259. https://doi.org/10.1201/9781003166702-14

Oladokun, V. O., & Oyewole, D. I. (2015). A Fuzzy Inference based Decision Support System for Solving the University-Course Admission Choice Problem. *International Journal of Computer Applications*, *112*(3), 1-7.

Pozzebon, J. A., Ashton, M. C., & Visser, B. A. (2014). Major changes, personality, ability, and congruence in the prediction of academic outcomes. *Journal of Career Assessments*, *22*(1), 75-88.

Rees Lewis, D. G., Easterday, M. W., Harburg, E., Gerber, E. M., & Riesbeck, C. K. (2018). Overcoming barriers between volunteer professionals advising project-based learning teams with regulation tools. *British Journal of Educational Technology*, *49*, 354-369. https://doi.org/10.1111/bjet.12550

Roushan, T., Chaki, D., Hasdak, O., Chowdhury, M., Annajiat, A, Rahman, M., & Arif, H. (2014). University course advising: Overcoming the challenges using decision support system. *16th International Conference on Computer and Information Technology* (ICCIT), 13-18.

Samaranayake, S., & Gunawardena, A. D. A. (2020). Dependency Evaluation and Visualization Tool for Systems Represented by a Directed Acyclic Graph. *International Journal of Advanced Computer Science and Applications*, *11*(7), 1-8. https://doi.org/10.14569/IJACSA.2020.0110701

Siddiqui, A. W., Raza, S. A., & Tariq, Z. M. (2018). A web-based group decision support system for academic term preparation. *Decision Support Systems*, *114*, 1-17. https://doi.org/10.1016/j.dss.2018.08.005

Soria, K. M., & Stebletin, M. (2013). Major decisions: motivations for selecting a major, satisfaction, and belonging. *NCADA journal*, *33*(2), 29-43.

Streufert, B. (2019). Advising Alternatives: A Case Study. *NACADA Review*, *1*(1), 14-29.

Wang, R., & Orr, J. E. (2019). Use of Data Analytics in Supporting the Advising of Undecided Students. *Journal of College Student Retention: Research, Theory & Practice*, *22*, 76-84. https://doi.org/10.1177/1521025119880819

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

Copyrights