Curri: A Curriculum Visualization System that Unifies Curricular Dependencies with Temporal Student Data

Dr. Stephen Michael MacNeil, University of California San Diego

Stephen's research focuses on how people collaboratively make sense of complex, 'wicked' problems. Wicked problems are dynamic and constantly changing. They involve multiple stakeholders, often with conflicting requirements. To address these challenges, Stephen develops sociotechnical systems that collect, organize, and use data to support reflection and collective action. He received his Ph.D. at UNC in Charlotte and is currently a postdoctoral researcher in the Design Lab at UC San Diego.

Dr. Mohsen M Dorodchi, UNC, Charlotte

Dr. Dorodchi has been teaching in the field of computing for over 30 years of which 20 years as educator. He has taught majority of the courses in the computer science and engineering curriculum over the past 20 years such as introductory programming, data structures, databases, software engineering, system programming, etc. He is involved in multiple NSF supported research projects including Learning and Predictive Analytics Research, Research Practitioner Partnership, Implementing Teaching Methods to help Students learn more efficiently in active learning, etc.

Erfan Al-Hossami, University of North Carolina at Charlotte

Erfan Al-Hossami is a Ph.D. student at UNC Charlotte. Erfan has been mentored in teaching CS1 since 2016 and then in CS education research. His work mainly focuses on predictive learning analytics. His research interests include Machine Learning, NLP, and Conversational A.I. and mental health. Recently, he's been learning more about code generation, transfer learning, and text classification.

Aileen Benedict, University of North Carolina at Charlotte

Aileen Benedict is a Ph.D. student and GAANN Fellow at UNC Charlotte, who has been mentored in teaching since 2016. Her work mainly focuses on CS education and learning analytics, with specific interests in reflective practices and predictive analytics. More recently, she has also been learning more about various topics in machine learning, recommender systems, and mental health.

Mr. Devansh Desai, UNC Charlotte

Devansh Desai is a recent graduate from UNC Charlotte working at Lowe's. He worked as a teaching assistant for two years, after which he focused on educational research. His research work mainly focuses on learning and predictive analytics and is experienced in data science and machine learning. Currently, he is seeking ways to contribute his time toward education and teaching.

Mohammad Javad Mahzoon, Core Compete, Inc

MJ Mahzoon received his PhD from the University of North Carolina at Charlotte where he also worked as a researcher to analyze student data. His research spans using data mining and machine learning tools to identify patterns, trends and anomalies in data.

Curri: A Curriculum Visualization System that Unifies Curricular Dependencies with Temporal Student Data

Stephen MacNeil, Mohsen Dorodchi, Erfan Al-Hossami,
Aileen Benedict, Devansh Desai, Mohammad Javad Mahzoon

Abstract

The correct sequence of courses in a curriculum can ensure that students develop their knowledge and skills holistically. The challenge level can also be more evenly distributed. Creating these sequences is difficult because curriculum designers must consider multiple potentially conflicting criteria simultaneously. There currently exists a dearth of tools for analyzing the curriculum that incorporates course dependencies as defined by curriculum designers while also considering students' pathways through the curriculum. In this paper, we present *Curri*, a data-driven curriculum visualization system that scrapes dependencies from our university's published curriculum and leverages student academic data to determine when, on average, students take each course. We evaluate our approach with a case study and two focus groups. This work provides initial evidence that considering both dependencies and students' temporal performance leads to new analyses and insights.

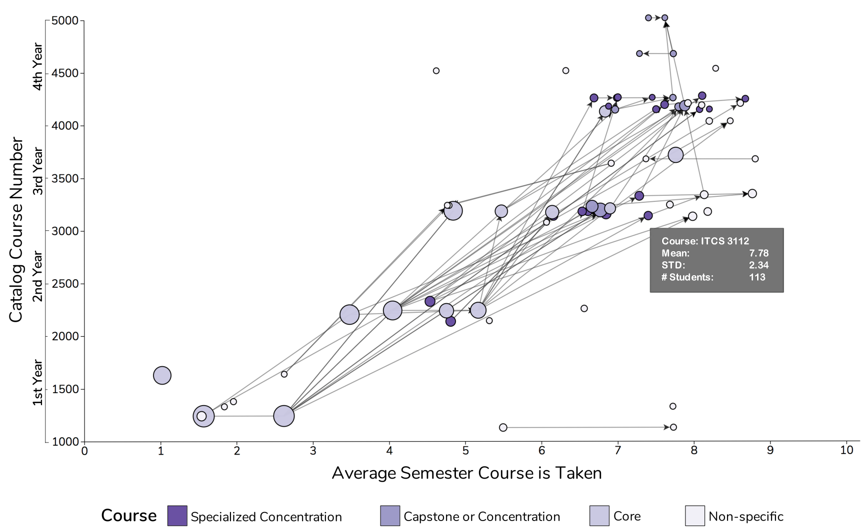

Figure 1: Curri, a curriculum visualization system that leverages curricular dependencies (pre-requisites) and student data to show students' pathways through the curriculum. Each course is represented by a circle, lines represent a dependency, and the placement of courses is based on when students take the course on average.

Introduction

The field of computer science education is continually evolving to keep up with new technology, improve on a lack of equity and inclusion[1], and to address the changing roles that computer scientists perform in industry[2]. Despite the adoption of new pedagogical initiatives and techniques [3,4,5,6] which bring aspects of collaboration, computational thinking, and equity to the fore; the curriculum has been the slowest to change[7]. Curricular change is a slow process due to many reasons. Curriculum designers often need to consider multiple, often conflicting, criteria. When redesigning an existing curriculum, designers consider the competencies that students need to achieve, a correct sequence of competencies to ensure preparedness, and the unique challenges faced by their students.

Accreditation boards and professional organizations; such as Association for Computing Machinery (ACM), Accreditation Board for Engineering and Technology (ABET), and Institute of Electrical and Electronics Engineers (IEEE); help curriculum designers by making curricular recommendations. These recommendations serve as important guides for curriculum designers; however, they don't take each individual university and student population into consideration. Curriculum designers often have to adapt these recommendations to meet the many complex needs of their specific student population. Currently, there exist many tools for visualizing the content and sequence of a curriculum[8,9,10,11,12,13,14]. However, none of these tools help curriculum designers determine whether a given curriculum meets the needs of a specific student population. Our work addresses this important gap in literature.

We present *Curri*, a curriculum visualization tool that incorporates curricular dependencies and temporal student data. Curri extracts curricular dependencies from published curricula online using web scrapers or as a file upload and incorporates students' course selection and performance retrieved from our student data portal. Like a good curry which balances heat and flavor, Curri balances students' actual pathways through the curriculum with the intended curricular dependencies. We present a series of case studies and focus groups from our college to illustrate the benefits afforded by a balanced view of the curriculum. For instance, multiple concentrations can be compared based on how spread they are across the curriculum and how many inter-concentration dependencies they have. Mismatches between curriculum designers' expectations and students' pathways through the curriculum can also be identified, such as curricular violations, where a course is taken very close to or before its own prerequisites. We conclude with a discussion about the effectiveness of our approach and our plan for future work.

Background

Representing the curriculum visually helps to communicate the structure, content, and sequence of the material. Ideally, this process ensures that competencies are sequenced correctly, that students have the freedom to create their own specializations, and that the amount of challenge is spread across the curriculum so that students aren't overwhelmed in one semester and under-challenged the next semester. These representations are often based on the curriculum designer's best estimates or based on an existing curriculum. These heuristics do not always correspond to students' actual pathways through the curriculum.

By plotting students' temporal pathways through the curriculum, it is possible to see where in the curriculum students are struggling most, to identify parts of the curriculum that students typically don't experience, such as an obscure elective, or to ensure that students are taking all of the courses necessary to ensure they have all of the intended learning outcomes and competencies embedded into the curriculum. Including this experiential information is important because while students theoretically take each course in the correct sequence, students may also obtain overrides or exemptions from the department. When many students receive the same types of exemptions it may indicate a problem, such as an improper sequence in the curriculum or a need for new or different classes. These views represent the student's pathway through the curriculum.

Curriculum: Design and Mapping

The curriculum guides teaching and learning. It not only "establishes the content for teaching"[15], but also identifies the "design specifications and constraints (student outcomes, competencies, learning goals)"[16]. Curriculum development focuses on the creation of these design specifications and constraints. It is further defined as the design, development, and implementation of guides for learning[8,17] such as topics, learning objectives, content knowledge, learning activities, and formal assessment[18]. Pratt defines the curriculum as "an organized set of formal educational and/or training intentions" which describes the curriculum as represented by the intentions of the curriculum designer[19]. Through this development, the curriculum itself is represented by the intentions of the curriculum designers. Curriculum guides the desired learning objectives, however it does not necessarily represent the actual applied curriculum[20] or the way that students experience it. Some suggest that curriculum design decisions are often based on "opinions, intuitions, and personal preferences"[16,21]. This highlights the importance of "a fairly accurate picture of the real curriculum"[20], constructed through evidence-based practices[21].

Curriculum mapping is one of such practices that can be used as a process to discover "what is actually being taught, how long it is being taught, and the match between what is being taught and [what is being assessed]"[15]. Auvinen et al. further describes it as relating course contents to programme-level goals. One example of performing curriculum mapping is to create a grid with courses as rows and programme-level goals as columns. A check could then be placed in each cell where the course contains that goal[8]. Two examples are: associating teaching and assessment methods aimed at identifying required "prerequisite concepts" and "assessment leaps"[18]; and comparing students' and instructors' perceptions of the curriculum[9]. By comparing the perceptions of multiple parties, such as both students and instructors, curriculum designers can then make better informed decisions and create curricula that is a more accurate portrayal of how students move through the curriculum.

Visualizing the Curriculum

Visual representations make it easier to obtain an overview and to explore the details of the data[22]. Additionally, the curriculum is a wicked problem with competing criteria that make it challenging to obtain an optimal solution [23,24]. Curriculum designers need to explore multiple possibilities and iteratively update the goals for the curriculum. For these reasons, visualizations are often used to represent the curriculum. Such representations, show the relationships between each course and suggest a sequence. While these intended sequences do not always capture the pathways students take through the curriculum in practice, they provide guideline for advisors to follow when suggesting courses to students. They evolve over time as curriculum designers continually improve the sequence of the courses and integrate new ones to address the needs of students and the demands of industry. In the following sections, we will see how the visual representation maps to aspects of curricular dependencies or student temporal data.

Graph-based representations

Graphs are often chosen for representing the curriculum because they match the structure of the curriculum which has courses and dependencies [8,9,10,11,12,13]. In graph-based representations of the curriculum, each course is represented by a vertex and each dependency is represented by an edge. Dependencies in this case often include co-requisites, pre-requisites, or both. These edges are directed and point from a source node to a target node where the target is the originator of the dependency and the source is the course that fulfills the dependency. Many graph-based representations use force-directed layouts to reduce clutter. In those cases, the resulting X and Y coordinates do not have meaning. It is also possible to choose coordinates for nodes based on a data feature; however, duplicate values for the feature will result in overlapping nodes.

Mapping curricular topics to courses or mapping between curricular topics is a common approach to ensure that dependencies are satisfied and that concepts are sequenced correctly to ensure that students are prepared at each point in the curriculum. This is often accomplished through directed graphs. STOPS is a system for matching courses and learning outcomes to model the curriculum. It visually presents the curriculum and learning outcomes as a graph to support curriculum development and planning [9]. Similar to the STOPS system, Siirtola et al. used a graph-based representation to map courses and topics together they also show how these map to other courses [10]. Rich et al. explored using graphs to represent dependencies between topics which could also inform curriculum design[12]. Based on their representation, material could be spaced out throughout the curriculum or throughout a course. Their model can help to ensure that students are prepared at each point throughout the curriculum but is general purpose and could be used to create a new curriculum without existing courses.

Analytics can also be performed on graph-based representations. Siirtola et al. generated a matrix representation of the distances between courses as measured by a graph of the curriculum[10]. Aldrich et al. also computed graph-based metrics for the curriculum including betweenness, centrality, and density measures[11]. Analytic approaches that build on a graph representation are scalable and can be used to analyze a curriculum that spans multiple majors or minors. This can be helpful for interdisciplinary students.

These graph-based representations are helpful to see related courses and to identify paths within the curriculum, but they don't often include temporal aspects such as when the courses are being taken by students. Some learning analytics frameworks consider the sequences of courses that students take to do risk classification[25]. These temporal aspects based on student data were missing from the graph-based representations that we surveyed.

Non-graph-based representations

Non-graph based representations include hierarchies[14], matrix-based representations[10,11], and trajectories that show how students progress through the curriculum[9,26]. Occasionally represented as graphs, hierarchies show hierarchical relationships between topics and courses. In curriculum visualization, hierarchical representations represent dependencies and show how each concept or course might be encapsulated in a broader structure such as another concept, a course, or a concentration. For instance, courses might be encapsulated within specializations or as in the case of Harden's curriculum map different biological topics such as hypertension are nested inside a more general topic such as cardiovascular systems[14].

Graph-based methods and hierarchies quickly provide an overview of the curriculum but they don't often consider the paths that students take through the curriculum (as measured by student learning data) or the temporal aspects such as when in their academic careers students are taking each course. Including student data is important because the intended curriculum is not always followed by students as prescribed. In our review we found two examples that consider the curriculum temporally. Trimm et al. show students' risks of not graduating over the course of the curriculum[26]. Plaza et al. compare instructor's expectations with the curriculum[9].

Spectrums and Dependencies

In this work, we build on our Spectrum and Dependency Graph model[27]. This model was used previously to represent the development of teamwork skills throughout the curriculum [27]. In that work, we presented spectrums and dependency graphs as a theoretical framework for considering both dependencies and temporal aspects. Though these two concepts are ideally related, there might be instances where most students encounter a badly sequenced set of teamwork experiences. The goal of spectrums and dependency graphs is to scaffold skill development temporally through the curriculum while making sure students are well prepared for increasingly complex teamwork. For instance, early classes help students to get comfortable collaborating, develop trust, and establish positive interdependence. In later classes, as the focus shifts toward performance and large projects, students should be well prepared to rely on each other and work as a team effectively.

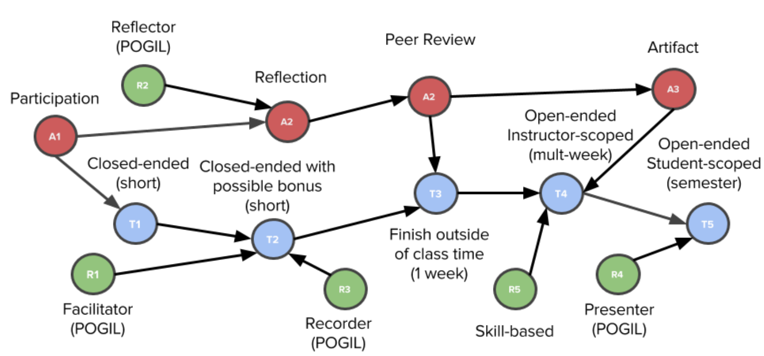

Figure 2: Dependency graphs show dependencies that exist between nodes.

Dependency graphs represent explicit intentions for how skills are related to each other. For instance, working on assignments outside of class requires students to budget their time, distribute tasks, and hold team members accountable. These dependencies ensure that teamwork skills are properly sequenced so that students are prepared as teamwork becomes more difficult and complex. By plotting dependencies and spectrums together, violations between these intentions and students' actual pathways through the curriculum can be investigated to ensure that teamwork skills and the type of teamwork is well sequenced. An example is shown in Figure 2.

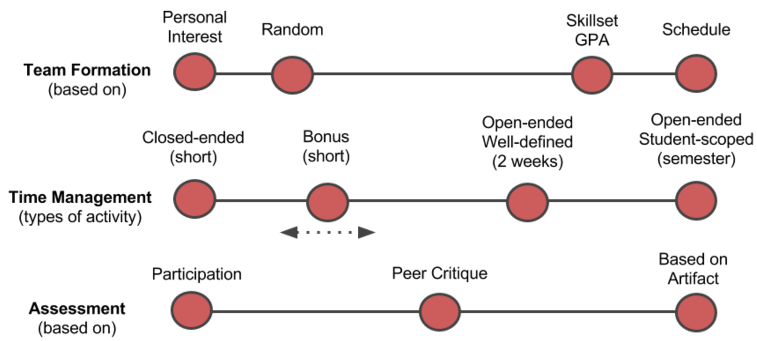

Figure 3: Spectrums place nodes temporally along a continuum to indicate when that pedagogical technique might be encountered by students.

Spectrums consider the temporal aspects of a progression. For instance, in introductory courses students might work together on low stakes tasks to become more comfortable working with others. Later in the curriculum, students may be expected to work in large groups on difficult projects with many tight deadlines. These spectrums, shown in Figure 3, only describe this temporal aspect of when a given type teamwork might be experienced by a student. For scaffolding skills, the temporal aspect might be determined by polling students across multiple grade-levels to determine what kinds of teamwork they are currently experiencing in their course. The teamwork type would then be plotted at the average perceived time in the curriculum. In this paper, we extend this theoretical framework to model courses within the curriculum.

Curri for Curriculum Visualization

We introduce *Curri*, a curriculum visualization tool inspired by the theoretical framework of spectrums and dependency graphs. Like a good curry which balances heat and flavor, Curri balances the intentions of curriculum designs as expressed through curricular dependencies with the actual pathways that students take through the curriculum as measured by course selection and course performance. Curricular dependencies include courses and their prerequisites which can be scraped from a department's website or a PDF. They can also be uploaded as a CSV file. Based on the dependency graphs metaphor, each course is represented by a node with dependencies between courses represented as directed edges. Students' pathways are captured in their transcripts, such as when they took a given course and their grade in that course. This data is aggregated across students for a single course to indicate when in the curriculum that course is typically taken. Leveraging the spectrum metaphor, each course is plotted as node temporally along a continuum based on the semester when students typically take that course. Curri overlays the dependency graph and spectrums, shown in Figure 4. This view shows both the curriculum

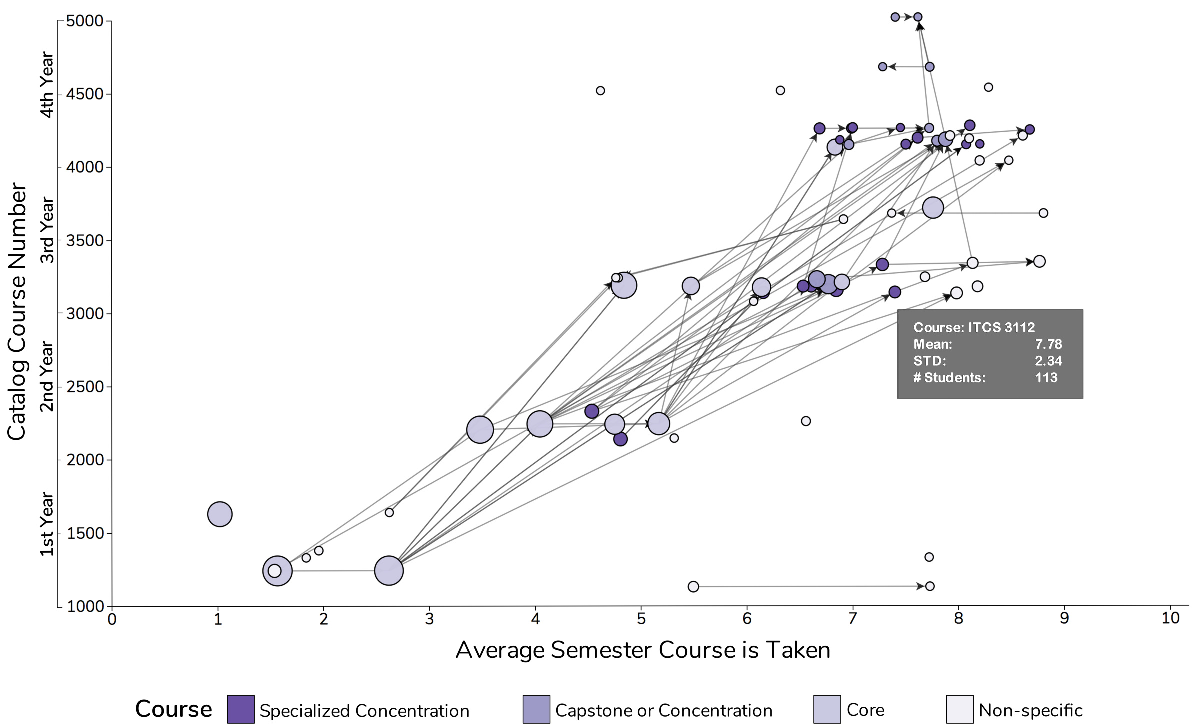

Figure 4: The curriculum is displayed as a dependency graph (nodes and edges). The X coordinate corresponds to the average semester when the course is taken and the Y coordinate corresponds to the course number. Color indicates course category and size indicates how many students took the course. The legend and axis titles were slightly edited in Photoshop for this paper.

designer's intention, as measured by the course number on the y-axis and the edges between courses which represents dependencies, and the students' pathways as measured by when the courses were taken on average, along the x-axis. Finally, the student data was filtered to keep only the maximum semester that an individual student took a course. This prevented students who took a course repeatedly from being counted multiple times when aggregating.

Curri was implemented using D3, a web-based visualization library, so that the visualization could be shared with administrators and faculty advisors. We implemented a few features to make navigation easier. Mousing over a node provides a tooltip with additional information, as shown in Figure 4. Clicking on a node highlights the dependencies, shown in Figure 6. We used three colors to categorize courses. We used grey for courses with no assigned concentration, light purple for the Core Courses (CS courses that are required for all concentrations), medium purple for concentration courses that are shared by more than one concentration, and dark purple for capstone courses and concentration specific courses.

As shown in Figure 5, the curriculum data is obtained using a web scraper. It is then formatted and combined with the student analytics data. We obtained our student data from our university; however, our university has uploaded student data to Multiple-institution Database for Investigating Engineering Longitudinal Development (MIDFIELD)[28]. In the future, we can obtain our data directly from MIDFIELD. By writing an additional parser for MIDFIELD, our approach can generalize to other MIDFIELD universities.

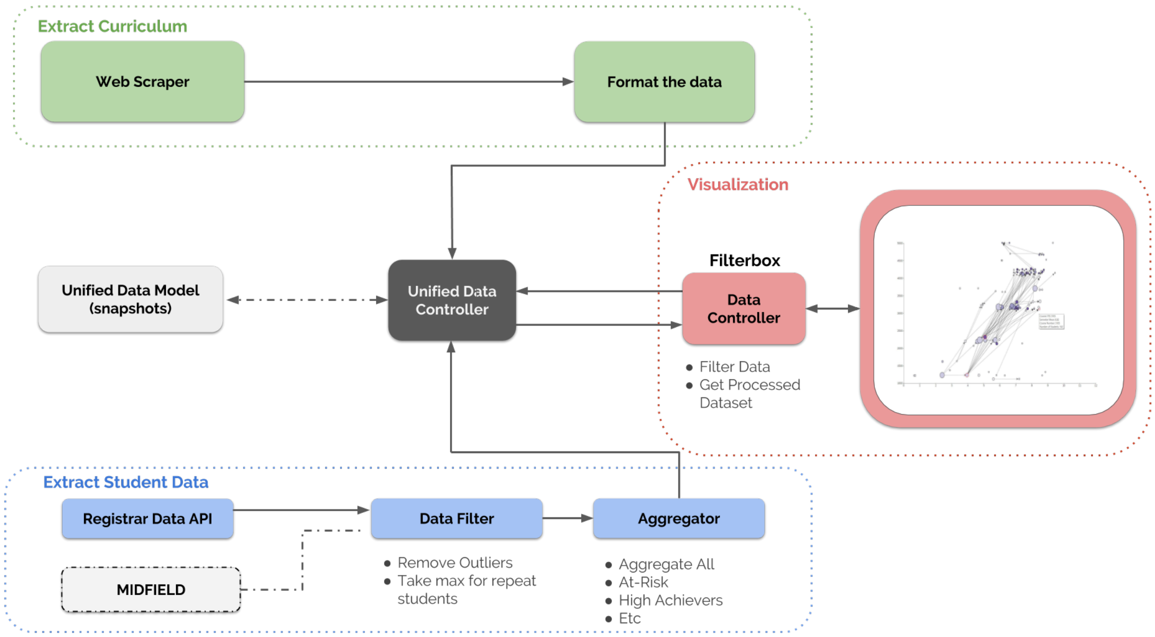

Figure 5: This block diagram shows how the data sources are integrated with the visualization.

A Case Study From Our College

The curricular data that was used in our analysis was obtained by scraping our University's 2015 course catalog. The student data included the academic records of 3605 students (population size) that took courses in our college between Fall 2013 and Fall 2016 (4 years or 13 semesters). This was the time period during which the 2015 course catalog was being used in our department. This data comprised every student that took a course in our college during this time period. This student data included course grades and the semester when the course was taken for each student. This data was obtained through our internal student data portal. This data was collected in accordance with an established IRB and was aggregated for each course before being sent to the web-app. The selection criteria for students that were included in the aggregation was students who started as freshman and excluded transfer students (resulting in 2,016 students included and 1589 transfer students excluded). The data for these 2,016 students (sample size) were presented in the visualization after being aggregated. The student data was aggregated for each course based on the average semester that a course was taken and on the average GPA for each course, and the standard deviation for both of these two measures. We did not have demographic information about students which might have included whether they were first generation or early-entry college students or information about their ethnicity, gender, or distance from home.

Constellations in our Curriculum

In this section, we present our results as constellations that we observed within our curriculum. We define a constellation as a set of related nodes and corresponding edges. The stellar constellation metaphor relates courses to stars and edges to lines. A constellation can be created by selecting a node or by applying a filter. To selecting a node, the user clicks on a node and then

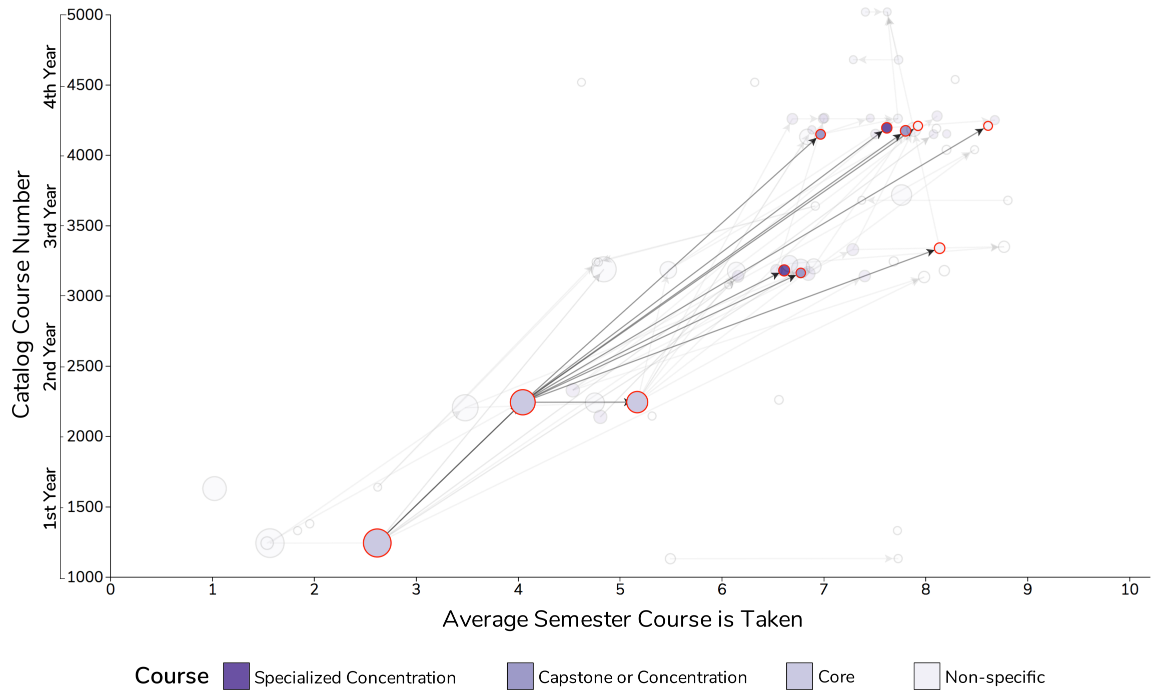

Figure 6: Clicking on a node (course) highlights connected nodes. Here, clicking on the *Intro to Programming II* node highlights the connected courses along with their dependencies.

all adjacent connected nodes are highlighted as well as their edges. We also created a filter for each concentration and for the core courses.

Below, we describe a series of constellations including the “Core Courses”, “Game Design and Development”, “Computing Systems”, “Capstone Courses”, and “Service Learning Courses”. These constellations can be compared by the overall shape, how stretched out they are temporally along the x-axis, and can be explored for back-edges which indicate that a prerequisite was taken before the course that it satisfies. These sequence violations are very quickly apparent in our visualization and other trends are also easily observed.

Constellation 1: Core Courses

As mentioned earlier, the core courses were colored with a medium purple. In Figure 7, the core courses are shown with other nodes filtered out. Core courses are the main courses that every student in the major is expected to take. As a result, these nodes are larger in the visualization because more students take them. Because these courses represent the foundational aspects of the curriculum, core courses are often spread across the curriculum and are very carefully sequenced through the use of prerequisites. This expectation is confirmed in Figure 7.

What was surprising about this constellation is that one of the core courses was taken very late in the curriculum (circled in red). By hovering on this node, we saw that the course is about the impact computers have on society. A topics course about ethics and trends in computing, students may take this course later in the curriculum to offset some of their more time consuming technical

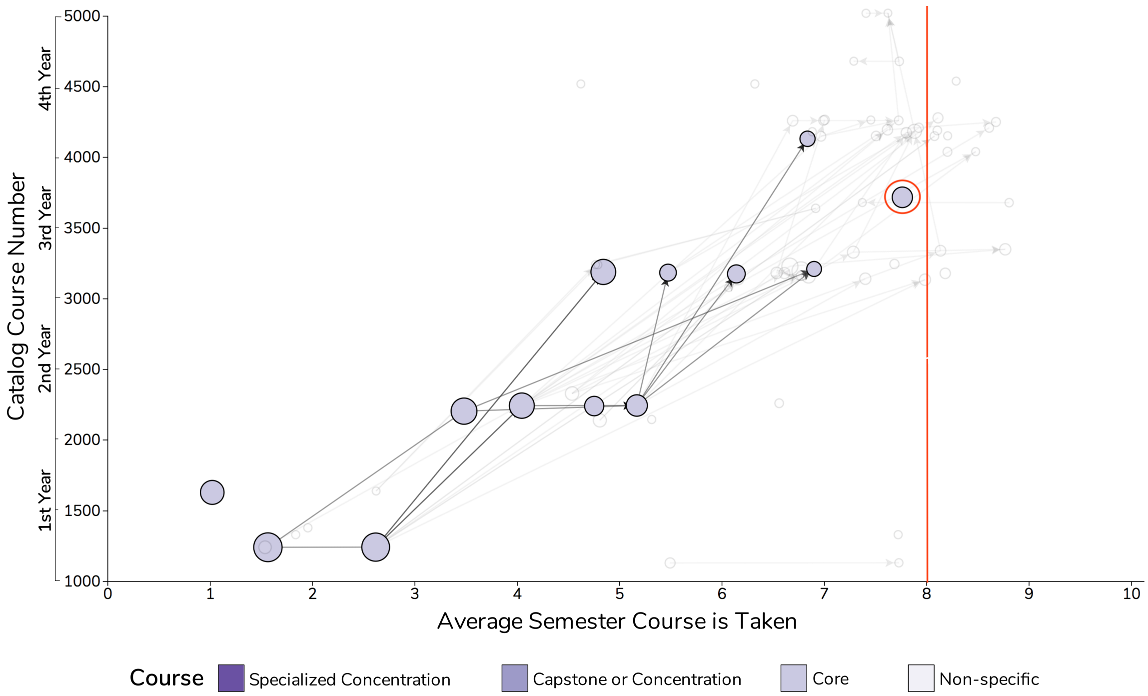

Figure 7: This constellation shows the core courses in our college. These courses are taken by every student to meet the degree requirements. A course is close to the 4-year mark.

courses. Based on the higher course number it is likely that curriculum designers intended for this course to be taken later to prepare students for industry; however, they may instead have intended it as a course to get students excited at the start of the program.

Constellation 2: Computing Systems

In our college, *Computing Systems* is considered a difficult concentration and it contains courses in advanced topics, such as compilers and operating systems. Shown in Figure 8, the courses are well spread out temporally along the x-axis suggesting that students don't often take multiple courses from this concentration at the same time. As a difficult concentration, it is not surprising that these courses are taken in different semesters to balance the student's workload in each semesters. It is also not surprising that some courses are taken on average in the 7th and 8th semesters of students' undergraduate programs. The dependencies for this concentration are mostly outside of the concentration with few inter-concentration dependencies. Having few inter-concentration dependencies can be beneficial because students often take their concentration later in their academic program at which point mistakes in sequencing can be more challenging to recover from while still graduating on time. Given the rigor of the courses in this concentration, we would consider this spacing and lack of inter-concentration dependencies the result of a well designed concentration. This is to be expected of a concentration that has been around for a long time and has been continually adjusted to meet student needs.

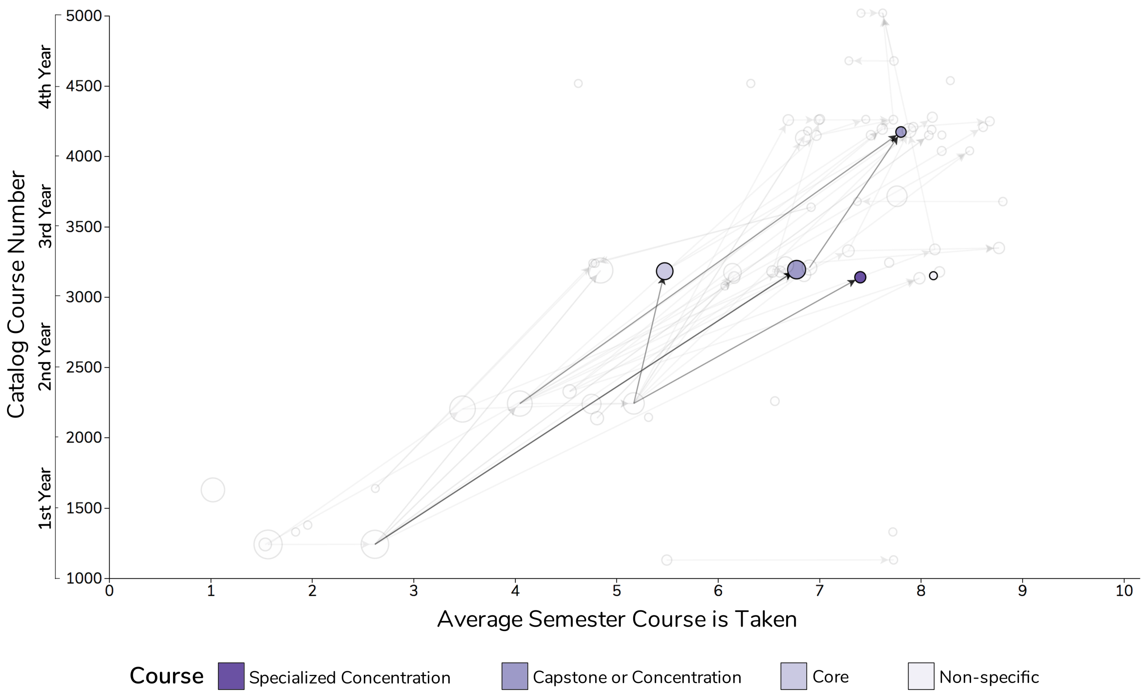

Figure 8: The *Computing Systems* concentration has few inter-concentration dependencies.

Constellation 3: Game Design and Development

The *Game Design and Development* concentration is a newer concentration in comparison to the *Computing Systems* concentration. It contains a mixture of classically difficult courses such as *Artificial Intelligence for Computer Games* and more design-oriented courses such as *Introduction to Game Design and Development*. Consequently, many of these courses can be taken at the same time without the student feeling overloaded. What was interesting about this constellation is that there are many inter-relationships between the concentration courses. These inter-relationships can make it challenging for students to plan their courses.

Constellations 4 and 5: Curriculum Violations

Curriculum violations are instances where a course is taken before its own prerequisites have been satisfied. This may indicate an inflexibility in the curriculum, a sequence that doesn't make sense, or an outdated dependency that has never been fixed. We found two violations in our curriculum, shown in Figure 10 and Figure 11.

In Figure 10, there is a violation related to a series of service learning courses. These service learning courses have an intended progression but in practice they are often taught in the same classroom by a single instructor. They vary only by workload, the number of credits earned, and the type of work that is expected. The courses are sequenced so that students first get experiencing doing outreach but then slowly move into a position of leadership. An inability to reliably fill the leadership course means that many students may violate this intention and move straight into a leadership position. It is also common for students to choose either version of the

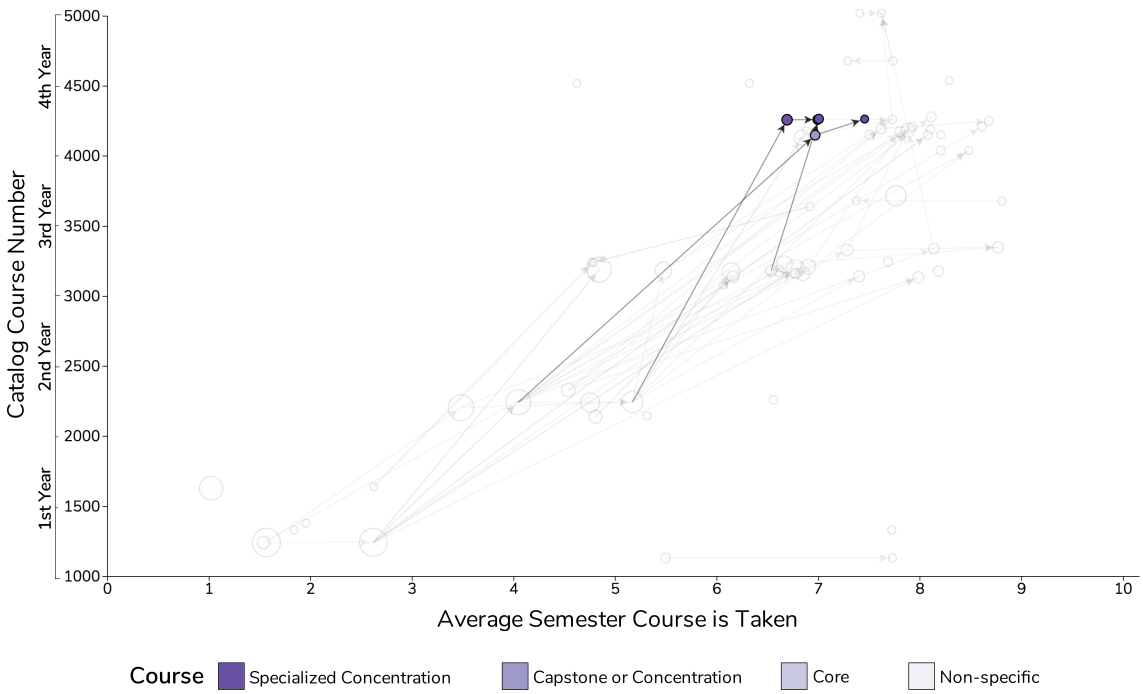

Figure 9: The *Game Design and Development* concentration is poorly spaced and has inter-concentration dependencies.

course based on how many credits they can afford to take that semester.

The second curricular violation that we identified is shown in Figure 11. This violation is the undergraduate thesis course. It has a prerequisite; however, students may become interested in doing the thesis at varying parts of the academic career, for instance when they start working on a research project with a faculty member. To encourage students to do research, this prerequisite is often overridden with the assumption that instructors will provide more assistance to students who are earlier in the curriculum. If instructors are uncomfortable supporting a student's thesis, they would obviously not approve the override.

Focus Groups with Instructors, Advisors, and Administrators

To better understand how to improve Curri for a department-wide deployment, we held two focus groups with 7 participants from our department. We recruited these participants through a snowball sampling method and announcements in faculty meetings. The inclusion criteria was that they hold a faculty, administrative, or advisor position in our department.The first focus group ran for approximately 90 minutes with 4 participants. Participants included one instructor, two advisors, and one who is both an advisor and instructor. They had seven years, three years, two years, and four and a half years of experience respectively for their roles. The second focus group ran for approximately 60 minutes with 3 participants. Participants included two administrators and one advisor with three years, two years, and three years of experience respectively. The focus groups were implemented as design studies to understand participants curricular needs and resulted in the identification of some challenges and opportunities associated with our tool.

Figure 10: Two violations of the curriculum. These service learning courses are frequently taken out of sequence.

Participants were given an opportunity to use Curri and then describe whether and how it might fit their needs. In subsequent sections, we analyze these focus groups using a grounded theory approach to identify and group themes based on what was said by participants.

Thematic Analysis of the Focus Groups

In the focus groups, we identified a series of challenges that administrators and instructors face when dealing with curriculum design. Based on their use of the tool, we also begin to observe some evidence of them using the Curri to address these challenges. The challenges that we observed aligned with three major themes: a lack of evidence associated with curricular decisions, difficulty staying up-to-date with an evolving curriculum, and varied needs based on their role in the department. We expand on these themes below.

A lack of evidence associated with curricular decisions was discussed frequently throughout the focus groups. Participants discussed how many decisions about the curriculum are currently being made based on a folk understanding of the curriculum. Participants were optimistic that Curri might be able to help them dispel myths and folk beliefs about the current curriculum. One participant described this benefit, saying:

> *"I think it's helpful and when I asked for 2175 ([a specific course]), it wasn't what I was expecting so I think it's really good... it would be a good use to dispel myths that sometimes sometimes get started that 'oh yeah students struggle because they don't take this early and you know.' " (P6)*

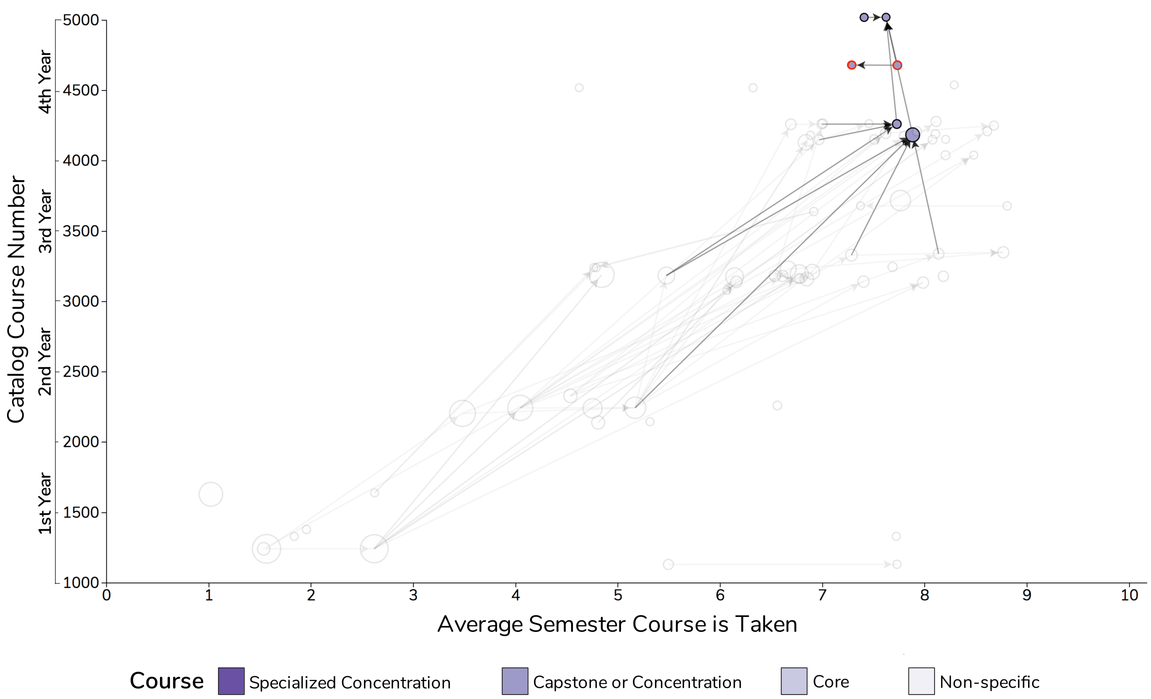

Figure 11: A curriculum violation: highlighting the capstone courses shows that the undergrad thesis is often taken early.

In this example, we see that P6 had beliefs about the curriculum and a discrepancy between their expectation and what they saw in the visualization caused them to realize they had a mistaken idea about that specific course. This benefit was described by other participants as well. P1 later added on that *"you actually have some evidence"* when using Curri. This evidence also enabled some insights from the participants. P3 also indicated that Curri could enable more evidence-based advising, saying, *"we can verbally say that based on anecdotal evidence, but then this [(Curri)] would give us and students historically have done this combination have gotten this semester GPA or whatever."*

Difficulty staying up-to-date with an evolving curriculum was a common theme that emerged in our focus groups. P6 indicated that specific curricular advice would not make sense *"post 2016, but [curricular changes in] 2015 does mess it up with like 2215 doesn't exist anymore."* These changes were hard to keep up with as P2 mentioned that her shifting role in the department has made it harder to keep up with curricular details, saying, *"Right. So yeah this may have been different. I used to know this curriculum. I don't anymore."* Curri addresses this challenge of frequent curricular changes by scraping the course catalog and student data on a daily basis. These updates appear to be very valuable to keep everyone on the same page.

The theme, *varied needs based on their role in the department*, was also expressed by multiple participants in our focus group. Participants often cited their background as an administrator or instructor to describe a specific goal that they needed to accomplish. Specifically, instructors appeared to be most focused on how to ensure that students met all the necessary requirements before taking their course, administrators hoped to identify bottlenecks in the curriculum, and advisors wanted to know which courses were complimentary and could be taken at the same time. P2 explained that as an administrator, they would like to identify bottlenecks and explore the underlying causes:

> *"Personally I guess this is more because of my administrative hat. I was thinking you know it'd be great to identify bottlenecks in the curriculum where our students are struggling the most and then try to see how to help them. Is it a problem with the curriculum? Is it a problem with the prerequisites? Is it a problem with some execution of the curriculum?" (P2)*

P4 described how their role as an advisor encouraged them to think about the load that students take on each semester, saying,

> *"For myself as a new advisor though I'm kind of focused on you know what our students taking together and being successful taking those things together. So that's it's nice to be able to see you know this four year graduates they took this you know set of classes together and they successfully move forward from that semester."(P4)*

Finally, instructors often talked about variations in how courses were taught and how these variations caused students to be unprepared for subsequent courses. For example, P6 said:

> *"I've run into issues where I'll keep it anonymous to faculty are teaching the same course just different sections and one faculty said I have absolutely no idea what the other faculty covers. So there's literally no discussion amongst the subtlety and no agreement no nothing so I have no idea if I don't know if the students talk amongst themselves and say take this course not that course you know which section which might run into some contribute to bottleneck issue.." (P6)*

Every participant in our focus indicated that they would use Curri again in the future, which is a testament to its ability to provide value across roles and serve as common ground for discussion. We developed Curri to provide an overview for instructors, administrators, and advisors. From the focus groups, we appear to have accomplished that goal with a variety of valuable discussions about the curriculum happening. However, this third theme suggests that more specific support should also be integrated for each of these specific role and task. As future work, we intend to add additional affordances to help identify bottlenecks and visualize the most common courses that are taken in each semester. We believe these additions will captalize on our initial success while also enabling deeper role-specific analyses.

Discussion and Future Work

This paper contributes the novel idea of combining curricular dependencies with aggregated student data. We present Curri, a curriculum visualization system which implements this concept and a series of case studies as constellations to evaluate the idea. Although a formal user study is still needed with faculty advisors and administrators, our case studies show promising results for how our system can be used to identify out-of-sequence courses or to compare different concentrations. We saw that more mature, difficult concentrations were well spaced with few inter-related courses to ensure students could progress through them and retake them if necessary without delaying their graduation.

By using both student data and dependencies, interesting new metrics, such as curriculum spread, can be used to compare different concentrations and clusters within the curriculum. Furthermore, graphs can be weighted using the distances between nodes as measured by the student data. These new distance measures might augment existing curricular distance measures, such as those used by Siirtola and colleagues[10]. Similarly, these new distance measures can be used for existing betweenness, centrality, and density measures of the curriculum computed by Aldrich and colleagues[11]. These graph traversals may enable curriculum designers to see whether some concentrations take longer than others. Until now, curriculum visualization tools have focused primarily on the dependencies that exist between courses or topics or both. We believe that these insights can lead to further development in this area. Consequently, this paper serves as a call to include empirical learner-centered metrics into the curriculum design process.

As future work, we plan to deploy Curri in our department for use by administrators and advisors. Before deploying Curri in our curriculum, we plan to add new affordances to detect bottlenecks in the curriculum by highlighting courses that are most frequently taken out of order or most likely to be repeated by students. We also plan extend our approach to other universities and departments. To extend our work, we plan to create a parser for MIDFIELD. This will enable other MIDFIELD universities to adopt Curri and explore their own curricula and student population. This extension will provide more information about the generalizability of Curri. To this end, our college has an annual enrollment of 300-500 students which means that outliers have less affect on the visualization. It will be interesting to see how robust Curri is for very small schools with few students. Finally, we plan to add more interactive features such as the ability to view different categories of students. By adding demographic and financial aid data, it would be possible to further slice and dice the data. For example, it would be possible to filter the visualization to show transfer students and other populations of interest such as students that are working part-time or full-time and students who are first-generation college students.

References

[1] Steve Cooper, Shuchi Grover, Mark Guzdial, and Beth Simon. A future for computing education research. *Commun. ACM*, 57(11):34–36, October 2014. ISSN 0001-0782. doi: 10.1145/2668899. URL `http://doi.acm.org/10.1145/2668899`.

[2] Susanne Hambrusch, Christoph Hoffmann, John T. Korb, Mark Haugan, and Antony L. Hosking. A multidisciplinary approach towards computational thinking for science majors. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, SIGCSE '09, pages 183–187, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-183-5. doi: 10.1145/1508865.1508931. URL `http://doi.acm.org/10.1145/1508865.1508931`.

[3] Mark Guzdial. Education: Paving the way for computational thinking. *Commun. ACM*, 51(8):25–27, August 2008. ISSN 0001-0782. doi: 10.1145/1378704.1378713. URL `http://doi.acm.org/10.1145/1378704.1378713`.

[4] Celine Latulipe, N. Bruce Long, and Carlos E. Seminario. Structuring flipped classes with lightweight teams and gamification. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, SIGCSE '15, pages 392–397, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-2966-8. doi: 10.1145/2676723.2677240. URL `http://doi.acm.org/10.1145/2676723.2677240`.

[5] Celine Latulipe, Stephen MacNeil, and Brian Thompson. Evolving a data structures class toward inclusive success. In *Frontiers in Education Conference (FIE 2018)*, pages 1–5. IEEE, 2018.

[6] Stephen MacNeil, Kyle Kiefer, Brian Thompson, Dev Takle, and Celine Latulipe. Ineqdetect: A visual analytics system to detect conversational inequality and support reflection during active learning. In *Proceedings of the ACM Conference on Global Computing Education*, pages 85–91, 2019.

[7] Steven Brint, Kristopher Proctor, Scott Patrick Murphy, Lori Turk-Bicakci, and Robert A Hanneman. General education models: Continuity and change in the us undergraduate curriculum, 1975–2000. *The Journal of Higher Education*, 80(6):605–642, 2009.

[8] Tapio Auvinen, Juha Paavola, and Juha Hartikainen. Stops: A graph-based study planning and curriculum development tool. In *Proceedings of the 14th Koli Calling International Conference on Computing Education Research*, Koli Calling '14, pages 25–34, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-3065-7. doi: 10.1145/2674683.2674689. URL `http://doi.acm.org/10.1145/2674683.2674689`.

[9] Cecilia M Plaza, JoLaine Reierson Draugalis, Marion K Slack, Grant H Skrepnek, and Karen Ann Sauer. Curriculum mapping in program assessment and evaluation. *American Journal of Pharmaceutical Education*, 71(2):20, 2007.

[10] Harri Siirtola, Kari-Jouko Raiha, and Veikko Surakka. Interactive curriculum visualization. In *Information Visualisation (IV), 2013 17th International Conference*, pages 108–117. IEEE, 2013.

[11] Preston R. Aldrich. The curriculum prerequisite network: Modeling the curriculum as a complex system. *Biochemistry and Molecular Biology Education*, 43(3):168–180. doi: 10.1002/bmb.20861.

[12] Kathryn M. Rich, Carla Strickland, T. Andrew Binkowski, Cheryl Moran, and Diana Franklin. K-8 learning trajectories derived from research literature: Sequence, repetition, conditionals. In *Proceedings of the 2017 ACM Conference on International Computing Education Research*, ICER '17, pages 182–190, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4968-0. doi: 10.1145/3105726.3106166. URL `http://doi.acm.org/10.1145/3105726.3106166`.

[13] Simone Kriglstein. Analysis of ontology visualization techniques for modular curricula. In *Symposium of the Austrian HCI and Usability Engineering Group*, pages 299–312. Springer, 2008.

[14] M Harden, Ronald. Amee guide no. 21: Curriculum mapping: a tool for transparent and authentic teaching and learning. *Medical teacher*, 23(2):123–137, 2001.

[15] Fenwick W English. Curriculum mapping. *Educational Leadership*, 37(7):558–59, 1980.

[16] Gonzalo Méndez, Xavier Ochoa, and Katherine Chiluiza. Techniques for data-driven curriculum analysis. In *Proceedings of the fourth international conference on learning analytics and knowledge*, pages 148–157, New York, NY, USA, 2014. ACM.

[17] Susan McKenney, Nienke Nieveen, and Jan Van den Akker. Computer support for curriculum developers: Cascade. *Educational technology research and development*, 50(4):25–35, 2002.

[18] Claudia Szabo and Katrina Falkner. Neo-piagetian theory as a guide to curriculum analysis. In *Proceedings of the 45th ACM technical symposium on Computer science education*, pages 115–120. ACM, 2014.

[19] David Pratt. *Curriculum, design, and development*. Wadsworth Pub Co, 1980.

[20] Fenwick W English. Quality control in curriculum development. 1978.

[21] Lori J Albert. Curriculum design: finding a balance. *The Journal of rheumatology*, 34(3):458–459, 2007.

[22] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages*, VL '96, pages 336–, Washington, DC, USA, 1996. IEEE Computer Society. ISBN 0-8186-7508-X. URL `http://dl.acm.org/citation.cfm?id=832277.834354`.

[23] Punya Mishra and Matthew J Koehler. Technological pedagogical content knowledge (tpck): Confronting the wicked problems of teaching with technology. In *Society for Information Technology & Teacher Education*

International Conference, pages 2214–2226. Association for the Advancement of Computing in Education (AACE), 2007.

[24] Katrin Becker. Wicked id: Conceptual framework for considering instructional design as a wicked problem. *Canadian Journal of Learning and Technology*, 33(1):n1, 2007.

[25] Mohammad Javad Mahzoon, Mary Lou Maher, Omar Eltayeby, Wenwen Dou, and Kazjon Grace. A sequence data model for analyzing temporal patterns of student data. *Journal of Learning Analytics*, 5(1):55–74, 2018.

[26] David Trimm, Penny Rheingans, et al. Visualizing student histories using clustering and composition. *IEEE Transactions on Visualization & Computer Graphics*, (12):2809–2818, 2012.

[27] Stephen MacNeil, Mohsen Dorodchi, and Nasrin Dehbozorgi. Using spectrums and dependency graphs to model progressions from introductory to capstone courses. In *Frontiers in Education Conference (FIE)*, pages 1–5. IEEE, 2017.

[28] Matthew W Ohland, Guili Zhang, Brian Thorndyke, and Timothy J Anderson. The creation of the multiple-institution database for investigating engineering longitudinal development (midfield). 2004.