Data-Informed Learning Design in a Computer Science Course

Daron Williams, Larry Cox II, Margaret Ellis, Bob Edmison, Taha Hassan, M. Aaron Bond, Quinn Warnick, Virginia Clark, Daniel Yaffe, Molly Domino, and Derek Haqq

Contents

Abstract

This case study chapter relates the experience of developing, evaluating, and iterating a media-rich asynchronous online course in light of increasingly sophisticated data-driven measures of course quality. The case will begin by introducing two guiding analytics frameworks: depth of use (DOU) and video analytics housed in the video content management system at the university. DOU is a measurement of the extent to which LMS tools and elements to promote learner engagement are incorporated into a course site, and it has shown a statistically

D. Williams (✉) · L. Cox II · M. Ellis · T. Hassan · M. Aaron Bond (✉) · Q. Warnick · V. Clark · D. Yaffe · M. Domino · D. Haqq
Virginia Tech, Blacksburg, VA, USA
e-mail: debo9@vt.edu; lacox@vt.edu; maellis1@vt.edu; taha@vt.edu; mabond@vt.edu; qwarnick@vt.edu; vclark@vt.edu; yaffedi@vt.edu; mollydomino@vt.edu; dhaqq@vt.edu

B. Edmison
Department of Computer Science, Virginia Tech, Blacksburg, VA, USA
e-mail: bedmison@vt.edu

J. M. Spector et al. (eds.), *Learning, Design, and Technology*,
https://doi.org/10.1007/978-3-319-17727-4_176-1

significant positive correlation with several indicators of student engagement, as well as with mean final student grade in a given course (Hassan et al., Depth of use: An empirical framework to help faculty gauge the relative impact of learning management systems tools. *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*. https://doi.org/10.1145/2543882.2543885, 2020). Video analytics are used to determine student viewing patterns and glean insights into how the instructional videos could be improved. Additionally, the case will explore the successes and challenges of designing an undergraduate computer science course to meet university priorities. This case will have implications for a wide variety of readers, including but not limited to administrators, professors, and instructional designers.

Keywords

Data-driven instructional design · Learning management system · Higher education · Computer science · Data analytics · Data-informed decision making · Systems thinking

Introduction

The instructional process is a system of interdependent elements: the learner, the instructor, the instructional materials, and the learning environment (Dick, Carey, & Carey, 2015). These elements interact in a dynamic and synergistic manner to achieve learning goals and objectives (Resier & Dempsey, 2007). Instructional designers and other faculty support professionals (from here on referred to as instructional designers) can help instructors design courses that better connect the interdependent elements for learning. Additionally, instructional designers are often uniquely positioned to influence organizational change through their work with faculty and organization leaders (Kowch, 2005). One such change is the growing use of data analytics to make pedagogical decisions in course design and instruction. Using data to make pedagogical decisions may have implications for student success.

Instructional designers and educational technologists can help instructors implement a data-informed decision-making process through thoughtful course design and content development. The learning management system (LMS) has become ubiquitous across the higher education landscape and allows for integrated data collection (Beck, Black, Dawson, DiPietro, & Jinks, 2007; Hassan et al., 2020; Samarawickrema & Stacey, 2007). A typical LMS can provide a treasure trove of data that can inform how we design effective courses (Leitner, Khalil, & Ebner, 2017; Mwalumbwe & Mtebe, 2017). A typical technology-enhanced course generates many data points that instructors can use to make instructional decisions. Instructional designers have a responsibility to learners to encourage course design decisions and revisions that improve the learning experience of students. Of course, instructional designers have less control over course decisions once the development process has ended. It is up to the class instructor to use interventions and the data collected from the LMS to influence real-time decisions and to decide how a course is revised in each iteration. Though many fields struggle to create courses that are responsive to student needs, the specific skill set being developed in computer science courses (i.e., coding, transferring between coding languages) make it hard for teachers to adapt instruction in the middle of a given unit (Guzdial, 2015). For such courses, the interaction between instructors and designers is key to data-informed course design and instruction.

An integrated approach to course design and development was recently employed at a large R1 institution in the United States. The university was asked to meet state and industry needs to graduate a significant number of computer science majors in a short timeframe. In order to meet the need for rapid course and program development, a team of instructional designers, developers, and faculty subject matter experts worked together to design computer science courses that use learning analytics for course correction. This case study will detail how a university-wide initiative to revise a program in computer science led to a data-informed design and teaching model to enhance the student learning experience.

Impact of New Degree Program/Larger Program Initiatives

In late 2019, lawmakers launched an initiative to dramatically increase the number of college graduates in computer science and related fields over the next 20 years. This "Tech Talent Investment Program" is designed to meet the needs of a rapidly growing technology sector in the state, which has created high demand for students with cloud computing expertise. This initiative will provide almost $1 billion to 11 universities over a two-decade period to expand bachelor's and master's degree programs in computer science, with funding tied directly to meeting enrollment and graduation targets at each participating university.

Our university is one of two institutions projected to enroll the greatest number of additional students in pursuit of these state-wide goals. As a result, the university has allocated significant resources to hiring additional computer science faculty, recruiting new students to our CS programs, developing curricular partnerships with community colleges and other 4-year institutions, creating a new professional master's degree, and redeveloping courses at the undergraduate and graduate level for delivery in online and hybrid formats. In early 2020, a group of sponsors and stakeholders from the Provost's Office, the College of Engineering, the Computer Science Department, and the learning technologies unit within the Division of Information Technology formally launched the "Computer Science Expansion Initiative" to rapidly scale up course development, upgrade technical platforms used to support CS courses, create self-assessment tools for new and potential students, and contribute to the university's recruitment efforts.

The university's CS Expansion Initiative included roughly three dozen individuals, organized as follows:

- A group of five sponsors who met monthly to provide high-level guidance to the project and allocate financial and personnel resources within their departments and offices.
- A 12-person steering committee made up of CS faculty members, instructional designers, and project managers, which met weekly to establish project priorities, determine the scope of work for each subproject, and hold participants accountable for project deliverables.
- Multiple project teams that met frequently (up to three times per week) to report on progress, clarify expectations, and coordinate work across various projects to ensure consistency and alignment with university goals. More than 20 people participated in the various project team meetings; however, the specific project described in this chapter included approximately 10 active participants.

Design Considerations

This course design and development project, centered on a 2000-level Computer Science course on "Data Structures" and intended to fill the role of "CS2," as it is known across academia, was one of the first projects in the broader initiative described above. It represented just one piece of the curricular puzzle, albeit perhaps *the* keystone of the entire initiative, given that the content from this course would be repurposed in several other parts of the initiative. So, while the following will detail the course design and development process for CS2 specifically, and descriptions of how data informed that design process, many other interconnected projects were simultaneously under development.

Data Frameworks

The need to collect and apply user data underpinned all of the design decisions made for this CS2 course. On the program side, the design team had to consider the need to create and market a site to prospective students, which allows them to self-assess their ability to apply basic CS concepts and helps guide them toward the best academic option for them. User performance data garnered by these interactions could also eventually be used to guide departmental and admissions decisions, though this has admittedly taken a backseat so far. But still, the need for some form of portability of assessment content and the data it may produce was never far from the forefront of the conversations guiding this overall initiative.

On the academic side, however, user performance data remained front and center for this project team for a more traditional purpose – to guide the team toward a good learner experience in the course and to identify shortcomings and suggest strategies to improve them. Two primary learning analytics frameworks played directly into this course: depth of use (DOU) and video analytics.

DOU Background

LMSs have provided an ever-expanding portfolio of services for educational institutions over the years. They have evolved significantly, in both the mode and scale of content development and delivery in the last two decades (Coates, James, & Baldwin, 2005; "6th Annual LMS Data Update,", 2018). A contemporary LMS, such as Instructure Canvas or Blackboard Learn, is a full-blown ecosystem of communication, productivity, and class-management applications and services. Understanding the degree and rate of adoption of LMS services is central, to university administrators, academics, and instructional designers, in improving the design, management, and delivery of course content. The diversity of pedagogies at an institution such as a state-chartered university, with thousands of courses offered in any given academic term, makes LMS adoption a fundamental metric of instructor's mode of interaction with students and with course content.

Qualitative reasons behind rapid LMS adoption have, therefore, received considerable research attention. A number of studies in the information systems (IS) and educational data mining communities have examined comprehensive measures of LMS adoption and success, taking into account system utilization, content quality, and student engagement and satisfaction (Adeyinka, 2011; Adeyinka & Mutula, 2010; Ozkan & Koseler, 2009). However, system utilization is often self-reported and monolithic rather than observed and specified as function of a course instrument or an LMS resource (Adeyinka, 2011). A key reason is that disambiguating the use of distinct resources from raw LMS data requires considerable effort towards web scraping, entity resolution, and summarization. This contributes to a lack of consensus on – and loss of generality for – techniques of measuring LMS utilization at scale.

This team's approach extends these studies by utilizing a course-level ordinal metric of LMS adoption called "depth-of-use" (DOU) (Hassan et al., 2020). Ordinal implies the metric takes the values of "low," "medium," or "high." The original authors investigated the relationship of DOU with course attributes like modality (audience, mode-of-delivery, third-party app use), participation (enrollment, viewership), logistics (number of teaching assistants, staff training), and outcomes (GPA, DFW) for a given collection of courses. This quantitative treatment allows researchers to examine, in detail, the key contexts of instructor engagement with and utilization of LMS services.

There is considerable prior work on qualitative grounds for LMS adoption, like teaching and learning efficiency, generational student expectations, and institutional expansion and consolidation (Berggren et al., 2005; Coates et al., 2005; West, Waddoups, & Graham, 2007). For course instructors, the basic predictors of the pace of LMS adoption are departmental affiliation (STEM vs. non-STEM, say) and course modality (online vs. face-to-face, say). West et al. conducted semi-structured interviews with 30 college instructors over two semesters, about primary use cases, teaching efficacy and efficiency, and overall satisfaction with Blackboard LMS (2007). The study identifies so-called "integration challenges": course instructors finding it difficult to integrate LMS services into their teaching practices. This notion of "integration" is echoed by McGill and Klobas for the case of student adoption of WebCT, whereby students with a more favorable view of the "task-technology fit" of LMS services are more likely to have higher LMS utilization (2009). The authors also note that instructor norms (instructor's view of LMS usability, support staff availability, and access to training resources) affect student utilization of LMS services favorably. Wilcox et al. surveyed user perceptions on frequent modes of use and platform limitations for Canvas LMS (2016). They identified a generation gap in expectations between students and instructors, wherein the pervasive student use of the mobile LMS app rendered a subset of course sites – designed by faculty members for the desktop – ineffective in navigation, flow and content organization.

Estimating Depth-of-Use

To measure the adoption of LMS features, the authors define a resource-level LMS DOU and describe how multiple resource DOUs can be aggregated into a single course-level DOU using simple logic rules (Fig. 1, see Hassan et al. (2020) for details of the aggregation strategy). They then tested nine hypotheses (Fig. 2) which evaluate how strongly DOU for a course is correlated with its modality, participation, logistics, and outcomes at the department level. For course modality, factors examined included audience (undergraduate vs. graduate), mode of delivery (online-only vs. face-to-face), and third-party app use. For course participation, enrollment and viewership (average weekly student-initiated page requests) were considered. For course logistics, the authors considered the number of teaching assistants and instructor's fluency in digital skills (as per participation in on-demand training coursework). For course outcomes, average GPA attained in the course and the DFW rate were considered.

DOU is a fairly recent construct, which has a goal of operationalizing the extent to which student engagement is designed into a course. To put it simply, DOU represents the extent to which an instructor makes use of the tools contained within the typical LMS which require students to actively *do* something, as opposed to passively delivering learning material via text or noninteractive multimedia (Table 1). Discussion boards, assignments, and quizzes are all common tools that are contained in a typical LMS, and each requires the student to engage with the content and produce something. Announcements do not typically require action or response within the tool, but they do serve as a conduit for interaction between professor and students. The use of each of these tools can be characterized as "low," "medium," or "high" by simply looking at an overall frequency of the number of times each one is used in a course.

Interpreting Depth-of-Use

According to Fig. 2, hypothesis **H1**, graduate CS courses have higher average DOUs relative to undergraduate courses (t-statistic is negative). Graduate courses make a significantly in-depth use of discussion forums while undergraduate courses have higher relative DOUs for assignment delivery and submission. According to hypothesis **H2**, online-only courses have superior overall DOUs relative to traditional in-class instruction. They are also linked to in-depth LMS use for online syllabi,

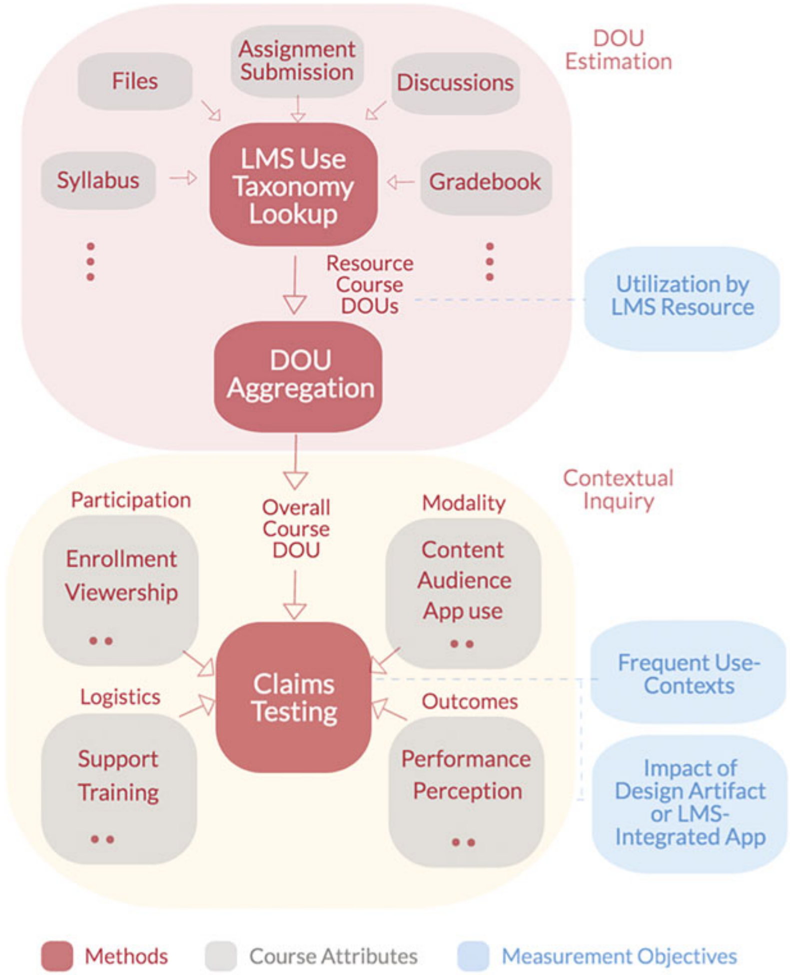

Fig. 1 Depth of use study visualized

announcements, gradebook, and discussion forums. **H3** reveals that third-party app use is linked to higher overall LMS DOU, with roughly 44% of high DOU CS courses relying on third-party apps. Higher DOU courses feature larger overall enrollment and viewership, per **H4 and H5**. Both of these are strong correlates of LMS utilization overall and across a number of LMS resources considered individually. High enrollment is linked to frequent use of detailed online syllabi, assignment delivery, and LMS gradebook use. While not linked to overall LMS DOU (**H6**), the number of teaching assistants is significantly linked to higher DOUs for assignment delivery and assignment submission. Higher enrollment in a broad-charter professional development program for digital skills is linked to higher overall LMS DOU (**H7**), as well as the use of announcements and gradebook. The average GPA attained

Hypothesis	H, p	F, p	t, p
H1: Undergrad.	23.4, <0.001*	24, <0.001*	-4.9, <0.001*
H2: Online	12.1, <0.001*	14.9, <0.001*	3.9, <0.001*
H3: App use	62.2, <0.001*	60.4, <0.001*	7.8, <0.001*
H4: Enrollment	30.6, <0.001*	20.9, <0.001*	-
H5: Viewership	783, <0.001*	72.7, <0.001*	-
H6: #TA	1.2, 0.54	1.4, 0.24	-
H7: Skills	8.3, <0.001*	7.5, <0.001*	2.7, <0.001*
H8: GPA	13.4, <0.001*	6.8, <0.001*	-
H9: DFW	10.0, <0.001*	5.8, <0.001*	-

*stat. significant, $\alpha = 0.05, F > F_{crit}$

	Ann	Syl	File	Asgn Del	Asgn Sub	Quiz Del	Quiz Sub	Gradebook	Disc Grp
Undergrad	1	1	1	1	1	0	0	1	1
Online	1	1	1	1	1	1	1	1	1
App Use	1	1	1	1	1	1	1	1	1
Class Size	0	1	1	1	1	0	0	1	0
Viewership	1	1	1	1	1	0	0	1	1
TA	0	0	1	1	1	0	0	0	0
Prof Dev	1	1	0	1	1	0	0	1	0
GPA	1	1	1	0	0	0	0	1	1
DFW	1	1	1	0	0	0	0	1	1

Fig. 2 Hypothesis tests to evaluate the relationship between depth of use and course modality, participation, logistics, and outcomes for VT computer science, overall (top) and by LMS resource (bottom)

in a course is significantly linked to overall DOU (**H8**) and the use of announcements, syllabi, and discussion forums. Course DFW rates (**H9**) are linked to overall LMS DOU, and the use of announcements, and gradebook.

DOU does have some limitations to note as it pertains to this design project. One, while it shows promise as a correlate of important student outcomes at the department level, DOU may benefit from continuous refinement of its source taxonomy. Two, at this moment, it only accounts for tools embedded directly within the LMS – does not take third-party technological tools, such as those integrated into an LMS via Learning Tools Interoperability (LTI), into account. If it did, it may have revealed even more insights about this CS2 course due to the course's heavy use of a third-party discussion platform. Three, DOU is currently based on simple frequency of

Table 1 Classification of Canvas LMS modes of use

Announcements (An)	Syllabus (S)	Files (F)
0: No announcements 1: A small number of placeholder announcements or schedules 2: At least one announcement per week or one per course-instrument	0: No syllabus 1: Syllabus under files, not embedded in canvas 2: Syllabus file linked and file previewed under syllabus 3: Syllabus embedded in canvas	0: No files 1: Nonzero meaningful course resources under files
Assignments: Delivery (A_d)	**Assignments: Submission (A_s)**	**Quizzes: Delivery (Q_d)**
0: No assignments on canvas or placeholder assignments without content 1: Assignments hosted on canvas, linking to DOC, ZIP, or third-party app 2: Assignments fully hosted on canvas	0: No file upload, likely paper or third-party application 1: File upload on canvas 2: Submission within canvas (text entry)	0: No quizzes hosted on canvas or placeholder quizzes 1: Quizzes hosted on canvas, linking to DOC, ZIP, or third-party app 2: Quizzes fully hosted on canvas
Quizzes: Submission (Q_s)	**Chat (C)**	**Discussions (D)**
0: No online submission, likely paper or third-party application 1: Submission within canvas	0: Chat disabled 1: Chat enabled but no activity 2: Chat enabled with activity	0: Discussions disabled 1: No discussion activity 2: Disc. Groups with activity

use, rather than considering *how* each tool is used. For instance, there are numerous use-cases of discussion forums in the LMS site of a course, and they can be a function of departmental precedents, course instructor's pedagogical preferences, and perceived burden of discovery and advancement. So, there is definitely more work to be done in developing a more nuanced view of stakeholder engagement from DOU-based platform analytics.

Video Analytics

The other primary source of data guiding this design project is video analytics. The video analytics natively available in the video Content Management System (CMS) can be used to help identify usage and video completion for the students in a given course. The analytics include completion rate, drop off rate, minutes viewed, and other useful metrics. The intent was to create videos based upon known best practices – that is, short, single-topic demonstrations when possible – and for the SME and teaching assistants to then monitor the analytics during the course and intervene if expectations were not met.

As is acknowledged in the Conclusions section, this course's starting position was relatively strong in relation to both DOU and Video Analytics, because the instructor had previously received training in effective online teaching, and she had a good understanding of what her mission should be in terms of creating quality videos for the course. That is, create videos of no more than 10 min in length, which demonstrate critical *processes* for effective implementation of data structures, and students should find value in them. There was no need to reinvent the wheel, but the design team still faced the considerable task of creating, editing, branding, and properly distributing over 100 such videos.

Project Team

Given that the needs and considerations going into this CS2 project were unique for this particular learning technologies organization at the time, it was necessary to assemble a project team that could shepherd the project to a successful conclusion. Especially, since the outcomes were defined by several sets of stakeholders. A single project manager (PM) was assigned to coordinate all elements of the larger initiative, and the PM played an integral role throughout. He attended most Subject Matter Expert (SME) meetings, as well as sponsor and steering committee meetings, to ensure that there was consistency and alignment between the goals of all the participating entities and interjecting with guiding questions anytime that alignment was unclear. The project manager also served as the central communication portal among all stakeholders and project team members.

Early on, the project team was larger than it perhaps needed to be, owing primarily to the novel nature of this initiative. In order to make sure that all needs would be met in this as-yet-undefined project, multiple instructional designers, web developers, multimedia developers, and university administrators were all at the table during the early stages of the project. Once a coordinated vision began to emerge, at that point the team could be winnowed down to allow for the organization to again allow for interference from other priorities. Some supporting instructional design and media development staffers were cleared to work primarily on other projects, with their assistance still being available as-needed. The final project team included the project manager, the primary SME and a few of her graduate assistants and colleagues, two instructional designers, one media developer, and two instructional design graduate assistants.

Organizational Design Considerations

Though detailed information was in short supply for the design team at the beginning of this project, there were a few bits of information to get us started. The initiative was specifically intended to support increased enrollment in Computer Science programs statewide at both the graduate and undergraduate level, primarily by making it easier for prospective students to find information about CS programs, indicate interest in programs, assess their own level of readiness, and then upon enrollment take the courses in a flexible manner that could work in the context of their lives, whether they intended to be dedicated full-time undergraduate students or working professional graduate students – or anything in between. To best serve the need for temporal and geographical flexibility for enrolled students, it was clear from the beginning that this project would need to culminate in a predominantly asynchronous online course.

To meet the outreach and initial enrollment needs, there would need to be a platform created to allow the prospective student to be aware of the program, to informally test their own readiness to join it, and to reach out to begin the enrollment process. This would require the creation of an open online platform to allow for these activities to occur and which could shuttle user performance data to the CS program for evaluation – a second layer of data-informed design unique to this initiative. This platform is a central feature of the overall expansion initiative, but a separate project team was dedicated to conceptualizing and building it; thus, it will not figure heavily into this chapter, aside from occasional mention due to its direct relationship to the CS2 content.

One critical piece that was clear from the beginning was that the content in a CS2 course could be a pivot point upon which a prospective student's level of readiness for masters' level work could be determined. Mastery of CS2 content is not just a second step in the base CS curriculum for undergraduate students, it is also considered a necessary prerequisite for incoming graduate students without a CS undergraduate degree or minor in CS. So the ability to apply these concepts could serve as the primary objective underpinning a readiness assessment. Demonstrate your ability to apply CS2 concepts, and you are at least minimally ready for graduate CS work. Fail to correctly apply them, and you could probably use at least a few undergraduate-level refresher courses first.

Finally, owing to one of the primary funding sources for this initiative being a state funding agency, any courses developed as part of this initiative should be able to be shared across public institutions in this state. Member institutions in this statewide public university consortium all use the same LMS, so while some considerations needed to be made when creating course content, especially multimedia content, almost anything built directly into the LMS could fairly simply be exported from one member institution to another.

Design Process

With the big structural considerations determined, it was time for the project team to dissect how to undertake this project. The team built an unpublished "mockup site" in the LMS, added the project team members, and began the project in earnest. With CS2 being a particularly content-laden course, the key problem to solve revolved around how best to arrange instructional content in a way that could meet the structural demands mentioned previously, remain "friendly" for SMEs and instructional designers to be able to pull and analyze data, and, most importantly, minimize cognitive load on the students. So rather than proceeding in chronological order with course content, the team decided to start by constructing one of the larger and more involved modules in its entirety. The team anticipated that this would help them visualize what "ideal" and "minimum viable product" might look like for a module in the course and would also provide a great starting point to get extra input and feedback from the SME, to ensure that the module would meet her needs, and then apply that to subsequent work.

The project team's work on this first module also provided the team with a great opportunity to determine how to best chunk up content in a way that would be clear and intuitive for students. There is no one "right" way to do this, as audience analysis and instructor preferences tend to play heavily into how a course is arranged. The selected approach ended up being a fairly straightforward structure that used the Modules tool in the LMS, populated with an introductory page for each module, and then the content pages with formative assessment checkpoints interspersed between some of them.

The primary SME and her colleagues were incorporated into the workflow early and often. As much as the rest of the design team was there to create materials and support her, the team all understood that she would carry a heavy workload in this process as well. She knew that she would have a lot of video content to create, some supported by a studio crew, some via screen capture technology on her own computer. Once the team determined how the content would be chunked and arranged, that allowed the SME to focus on scripting and creating her videos and determining the sequence of pages and assessments in the course. As she did this, she kept detailed "readme" documents for each module.

Readme documents are commonly found when installing computer software. These documents provide a general overview of the software, suggested instructions for installing the software, and potential issues that could come up when installing the software. Some versions of readme documents can contain more information, but the above is a list of items that can usually be found. Since the SME was a computer scientist, it made sense that this approach was adopted to explain how to "install" the module content in the LMS. These course module readme documents contained the module content, practice assignments, assessment, notation for images to be used in the module or in the assessments, and the sequencing of the content. The team could easily open one of these documents and know how to structure the material in the LMS. Of course there would be items that needed to be corrected, but having these documents cut down on the amount of meeting time needed to finalize a module. Additionally, reporting progress via the completion of the readme documents made it so that it was always clear where progress stood with regard to any particular module.

The lessons in most courses in the program will contain a page that introduces a topic, usually via a brief (2 min or less) stand-up style video of the instructor, and which contains associated resources such as presentation files. After that comes content pages with fairly similar layouts to deliver the material and interspersed practice opportunities and assessments wherever possible. This particular CS2 course served as the "canary in the coal mine" in terms of figuring out the important familiar elements to include and how to sequence them, and the team has subsequently created more courses using this one as the desired archetype.

Course Details

The content pages in each module typically each included a short topical video usually 8 min or less in length, a corresponding presentation file or associated resource, and embedded coding practice opportunities where applicable. The checkpoints were ungraded quizzes that students could use to monitor their learning progress. Each module, or each section of a module in the case of the more complicated modules, culminated in a graded Section Quiz.

Regarding these interspersed assessments, one of the CS faculty members alerted the design team to the existence of several question banks related to introductory CS topics after learning about them at a conference. These question banks, known to us as the “Canterbury Questions,” after the conference where they were proposed in Canterbury, England, were created via a joint effort by many professors in the CS community (Sanders et al., 2013). The idea was to build up a collection of questions about the various topics that may be covered in a CS1 or CS2 course. One could then pull these banks and questions into their course to use. These banks were in a format that allowed them to be imported into the LMS.

Due to the importance of modularity of content to the overall initiative, these modules were arranged by topic rather than by week of instruction, which would allow the team to pull content into other environments such as the assessment platform quickly and easily and would allow the SME flexibility to deliver the course across a broad variety of time domains (i.e., 16-week semesters, 6-week Summer semesters, 3-week Winter terms, etc.) without having to fundamentally alter the content. This institution’s instructional designers have found this to be a generally good practice anytime a course is regularly offered outside the typical Spring and Fall semesters, as it allows SMEs to just pick the existing topics for a given week of instruction and communicate that to the students, rather than having to actually make changes to the underlying content itself, as would happen if materials are arranged by day or week.

While introductory pages were included at the beginning of each module as a self-regulatory aide for students, to help them stay connected to the bigger picture of the module and course, the team decided to also include a navigational aide at the top of each content page. The LMS’s embedded “Previous” and “Next” buttons could easily shuttle students forward or backward one resource at a time, as determined by their order within the Modules tool, but this took it a step further by including hyperlinks to every page in a given section, so that students could navigate to *any* other page related to that topic, at any time.

To zoom back out a bit, the process of developing this first module, from first conceptualization to “98% completion” (because, of course, nothing is ever 100% done in our field!), took approximately 1.5 months, even with a whole team devoted to it and an extremely responsive SME. However, many of the broader structural decisions and a few rounds of SME feedback happened in the first 3 weeks of that time, so the team was able to begin some work on other modules prior to its completion, thus beginning the more “routine” business of developing multiple modules concurrently. As mentioned early on, though it is often the case that it takes significant effort at the beginning of a design project to establish momentum, suffice it to say that the unique demands of this larger initiative, including the data-related demands of how to build and organize a course in this context, added to the difficulty of overcoming the “object at rest” inertia of this course design project.

Once the general structure and flow of the CS2 course had been determined, course development continued in a predominantly chronological fashion. Though the project was slated to “go live” in the Fall 2020 semester, the team realized there was an opportunity to pilot some material in a Summer offering of the course first, so the team focused on getting a few particular modules ready for that purpose, and three complete topical modules were released inside the Summer 2020 site. The instructor accompanied these modules with a feedback opportunity for the students. This allowed the SME and the design team to get some insight into what might prove more or less valuable for the students, while there was still time to incorporate those insights into the fully online course for the Fall.

Challenges

There were several challenges that the team met over the course of this design project. Some were relatively common challenges that any design project will encounter, such as proper alignment with the academic program’s curriculum, how to best serve this unique audience of students, etc. Others, however, were new and emergent challenges that the team had to work through in the moment.

Getting Started

The initial challenge was simply overcoming the “inertia” of the project. There had been many high-level conversations between the computer science stakeholders and the design group’s higher-ups in the months leading up to the project, but important details seemed to change regularly, and though the minds were moving quickly, the hands had yet to hit the keyboard. Even when the project was supposed to officially kick off, it still seemed as if central details were less concrete than expected. But as instructional designers everywhere can likely identify with, there comes a point at which you just start moving and deal with the changes as they come, so the team proceeded with this in mind.

As mentioned before, just figuring out who should be in the room was a challenge at first, but fortunately the larger initiative had garnered enough attention as an important component of a major expansion of the university that the involved departments were willing to devote human resources to it. The team ended up having more support than was actually needed, so while work was distributed to all participants initially, it became clear after a few weeks that the team could handle the necessary work more efficiently. There was support “on retainer” in the form of knowledgeable media developers if the workload were to increase unexpectedly, but for the most part the team was able to move forward with the streamlined group described earlier.

Establishing a workflow provided some difficulty at first as well, because while the instructional design group was already well-versed in “standard” course design projects, this project involved support personnel from multiple departments, from the CS program, and from the College of Engineering which houses CS at this university, and others from noninstructional-design areas of the learning technology organization. Figuring out how to incorporate everyone effectively was another piece of the initial puzzle, but eventually some of these support personnel became the on-retainer support crew, available in case of emergency.

The unique circumstances brought on by COVID-19 also affected this workflow. Colleagues who could have otherwise chatted at the proverbial water cooler had to instead learn to use a chat platform which usually (but not always) led to relatively quick answers. Communication with the SME and her team would have previously happened primarily via regular face to face meetings but had to shift to a web conferencing platform and cloud-based shared documents. One element that was particularly helpful was the SME’s “readme” documents for each module. Though these are ubiquitous in the CS world, this instructional design group had not previously made regular use of them. As a result of this project, readme documents have since factored into multiple other CS course design projects and may likely remain part of the university’s standard instructional design practice.

Assessment Challenges

The next challenge was to create the course in a manner in which assessment content could quickly and easily be exchanged or pulled into the self-assessment platform also under development for the Computer Science program. The design team made the determination that, regardless of the eventual destination of assessment content for this platform, housing course assessments directly in the LMS would likely give us the flexibility and enough portability that the project team working on the assessment platform could make efficient use of it. This conceptual separation of the course assessment content from the freestanding assessment platform allowed us to finally move forward confidently with the course design project.

However, there were some formatting issues with the questions that had to be addressed. This predominantly boiled down to editing the coding examples so that they looked like code and contained the correct syntax. The other need was to correct the images in the questions, add alt text to them, and resize them so that they displayed correctly. In addition, to make sure that these banks would be useful to subsequent classes, a separate site in the LMS was used to house the questions so that they could easily be imported to another course.

Accessibility Challenges

Accessibility of online course content is of course a critical consideration for any project, and this one was no different. With a big part of the mandate of the overall initiative being the ability to reach a larger audience, the team needed to ensure that all content, regardless of how “content-heavy” any course might be, must be as flexible and accessible as possible. That means all presentation files are properly designed with regard to alt-text for all images, all text built in or housed in the LMS makes proper use of Headings, all fonts are appropriate for the delivery mode, and all video content is captioned and/or transcripted.

Particularly of note in this realm is that all of the 120+ videos needed to be captioned. While this sounds conceptually simple enough, the details of video “ownership” inside the video CMS complicated matters. The SME would create a video, whether in a studio environment or with a screen capture tool such as Camtasia, and the media developer would add branding and design elements to ensure a consistent feel throughout the course, make the necessary edits to the videos themselves, and then upload the videos to the video CMS. This means that, according to the video CMS, the media developer is the “owner” of the videos. As a result, the media developer had to manipulate several settings in each and every video to allow access for others to view analytics, request and edit captions, and so forth. It was a considerable burden on that person, and though the team did establish a system by which to request professional captioning for all videos, all requests had to be made at least a week ahead of time in order to return in time for the video’s release. This was no problem with the majority of the videos, but time did creep up such that the team was no longer comfortably several weeks ahead of each module’s “live” release by the time the course’s final videos were finished.

One final accessibility challenge involved the embedded CS digital textbook and coding practice environment. The digital textbook was developed over a decade ago by CS faculty and had not received recent accessibility-related updates and remediation. To accomplish this would be to wade a bit farther into the deep end of the CS pool than our instructional designers were prepared to go. Fortunately, this accessibility remediation was something that had been in the queue for our CS faculty for some time, so this initiative provided the perfect time and opportunity for them to address this need, and it has since been completed.

LMS Challenges

A couple of LMS-specific challenges also emerged as the course developed. The team provided custom navigation in the form of hyperlinks to all other pages in a particular Module or Section on each content page in the course. This turned out to be a larger challenge than expected, for a few reasons. First, if a design team engages in creating something like this, we can now confidently say it should be among the *final* steps in the hands-on course design process. The team can map out the order of pages and assessments in a course, but inevitably plans will change here and there, and one simple change in the page order means that *every* page in a particular module or section will require updating the order of the hyperlinks. The design team did learn this lesson the hard way. Also, changing the name of a page will break any hyperlinks that were already created to direct students to that page. So each change of that nature also requires subsequent hyperlink management.

Another LMS-specific challenge came in the form of embedding videos inside the content pages. The good news here is that the LMS and video CMS do “connect” via a button in the LMS’s rich content editor, so embedding an existing video in a content page is functionally as simple as clicking a button, finding the right video, and clicking a confirmation button. However, due to ongoing updates to the video

CMS, simply placing a video this way leaves a large blank space directly below the video. This can interrupt the flow of any page in which a video is followed by more content or resources.

The solution here was twofold: first, include a short instructional sentence prior to each video, such as "*Watch the video and view the associated slides, then click 'Next' at the bottom of the page to proceed to the next activity.*" This is a small cue, but the words are brief, specific, and descriptive, such that a student can be expected to understand that there is more to do after watching the video. Second, the instructional designers did make a minor HTML change to the size of every video's iframe, which reduced the height by 135 pixels, from 775 to 640, while still allowing the transcript to show up in full and with unaffected functionality. This adjustment is not likely a service the group will continue to offer most clients.

Learning the Hard Way

The final category of challenges combines many of the previous challenges, and it happens at the junction of learning several aforementioned lessons "the hard way" and migrating the content from the unpublished development site to its final home in the actual semester-associated course. Perhaps the key difficulty here happened with the formative in-module assessments, as all of the hard work to set up assessment questions in the development site was essentially for naught once the content was migrated, due to the fact that question banks could not be migrated directly without manual intervention in each and every checkpoint quiz. Fortunately, upon learning earlier in the development process that the question banks did not move from place to place quickly and easily, the team did control for this fairly well by documenting the specific steps taken for each question in the development site, which made re-doing the work in the final site much easier.

Conclusions

The design team reached several conclusions regarding the process of designing a data-informed course to effectively meet the needs of a broad variety of stakeholders. Proper use of existing data, forming an effective team, and creating processes and documentation to improve efficiency in the future are all concrete ways this design process impacted this team and the organization.

Data Revelations

Regarding DOU data, this particular iteration of the course was already starting with a DOU rating of medium. Part of this may be due to the course having already gone through a redesign with the assistance of an instructional designer a few years prior – this redesign took the course from a standard face to face format to a still live, but more technology-enhanced format, with enhanced assessments and embedded practice opportunities. However, this second redesign was still able to raise the level of

DOU for the course; Syllabus, Quizzes: Delivery, Quizzes: Submission, Assignment: Delivery, Assignment: Submission, and Files were all rated high, but Announcements was raised to high from the redesign. This increase was able to bring the overall DOU rating for the course to "high." Again, while this does not tell us about the quality of the implementation, it does tell us that there is likely high student engagement in the course due to the high utilization of the LMS.

In practice, the postredesign video analytics data did not prove as impactful as expected. However, it is worth noting that this may be a very positive sign in this particular case. The primary SME for the CS2 course possesses a unique combination of prior training in instructional design and a very good sense for her students' needs. The SME had worked with the instructional design group to develop a prior version of this course, offered in a different modality, so she was already aware of the best practices of video creation. Not only were the videos generally created to high standards from the start, but the pre-release of a portion of the new asynchronous content in the condensed Summer version of the course proved to be a very informative data collection point.

Specifically, this pre-release of videos allowed the team to diagnose a shortcoming in the videos as they existed at that time: There was no speed control for the videos, which would have allowed students to watch the videos at a speed between 0.5x and 2x the original speed. This feature gives students flexibility and control over their viewing experience, and this feature is basically considered essential now, given that it is a standard-issue feature on most familiar video streaming platforms. Releasing the asynchronous video content in small chunks to the Summer students, and giving them the opportunity to review and reflect upon their experience, gave us not only the awareness but also the time to investigate whether this was something we could address prior to the official release of the full course in the Fall. As it turned out, though it required a couple of weeks of discussions and negotiations between video CMS administrators on campus, the team was able to get this feature included for all videos housed in the CMS, not just the ones included in this CS2 course.

That said, mid-semester analyses of the video analytics data were not completely fruitless. The data did not provide much in terms of actionable insights, but they did at least provide a glimpse into how students were consuming these videos. Perhaps predictably, the data showed a bit of a bell curve – a small number of students faithfully completed every video, a small number of students essentially watched none of the videos, and the majority of students fell in between. This is essentially okay in a course like this, given that the primary point of the course is that students can apply these concepts – if some students are well-versed in data structures, they may not feel the need to consume video content. So though actionable insights from video analytics were sparse, the process did serve to confirm that viewing patterns seemed to follow expectations.

The other primary insight from mid-semester spot checks of video analytics is that, perhaps predictably, video viewership spiked in the days preceding major assessment dates. This is perhaps unsurprising, but it does reinforce the value of creating asynchronous videos not only for student use while learning about particular topics, but also while reviewing those topics prior to a test or leading up to a project.

Regarding video analytics, the instructors and teaching assistants were provided professional development on how to access and use the video analytics. As analytics were used, some problems became evident. Some features of the analytics were difficult to navigate for various reasons, including user interface, verbiage, and the fact that not all analytics were complete. One major problem was that users could view a video without being logged in. This meant that the analytics had an “Unknown User” that was an umbrella entry for many views.

Another aspect that could be a problem in the future was that videos would need to be copied from semester to semester or from section to section, so it does require extra effort to filter out the data for users from previous semesters or other sections. The last aspect that was troublesome was the manual assignment of access to the analytics for all of the videos. Teaching assistants need to be manually added to each video, which becomes cumbersome to manage in a course with over 100 videos. New teaching assistants will also need to be added to previous copies if you want to compare historic trends.

Design Revelations

One major takeaway might be that the likelihood of data-centric design projects feeling overly complicated and impossible to get started is high, but once you have clarified the parameters and necessities surrounding course and user data, establishing forward momentum is possible by focusing on one learning module, establishing what the minimum viable product might look like, and proceeding from there. The design team chose a specific initial module because of two characteristics: The first was that it was more complicated than the other topics, and the second was that the SME was very familiar and comfortable with the content of that module. It only took a couple of days of focus for the SME to figure out how she thought the chunking should go for that module, and that provided all the team needed to really get moving. So though progress felt impossible at first, taking this one concrete step allowed the SME and the design team to move forward, and what was initially conceptualized as the minimum viable product ended up blossoming into a clear and useful structure upon which to base further content.

The practical impact of this initial difficulty may be that it exposed one of the most substantive takeaways for the instructional design team: It laid bare a need for more structured intake processes and documents and impressed upon the team and their supervisors the importance of direct involvement as early as possible in any discussions surrounding program redesigns. The instructional design team has since been able to leverage this knowledge to join other academic programs in their curricular discussions that include the teaching faculty as well as departmental decision-makers, and it has made a stark impact on the clarity of vision and direction the group can bring to the design process. Put simply: If at all possible, in any data-centric program redesign situation, the earlier the instructional design team can join *and affect* the program-level conversation, the better.

As alluded to in a previous section, the design team benefited greatly from the usually CS-specific concept of readme documents. This sparked a broader movement by the instructional designers toward better data management and sharing. Setting up a shared drive or folder to serve as the central hub for designers and SMEs is a good start, but providing extra structure and context inside those shared documents, such that SMEs know *what* to share and *how* to share it, goes a long way toward providing valuable and time-saving clarity for the design team. Instructional designers know the value of utilizing technological affordances, job aides, and advance organizers in ensuring that critical knowledge does not slip when it is needed most. This situation presented a golden opportunity to gently guide SMEs toward organically providing the right information, clearly, in a way that allows the design team to pick it up and run with it when ready.

Another key takeaway might be to use the surrounding circumstances to your advantage. Though the team had about 4 months from establishing initial forward momentum to the course's go-live date, and though we may have sprouted some gray hairs rushing to get some content developed for test release during the Summer course offering, well earlier than the initial course deadline, it was totally worth the effort. The feedback from the Summer students helped us identify a small but impactful shortcoming (the lack of speed control on videos) and gave the team time to figure out that the issue could be fixed at scale, not just for this project, but for the benefit of the entire university. There may be other useful resources at your disposal, perhaps as simple as chatting informally with faculty or colleagues who have subject matter expertise. Use anything and everything you can, because you never know what tiny revelation might make a big impact.

A final design-specific takeaway is to take good notes. As the design team worked through the challenges this project offered, all challenges and solutions to those challenges were faithfully documented. Upon completion of the project, once the team had time to reflect on the process, these solutions, as well as the things that we wish would have been addressed up front, were incorporated into the institution's instructional design team's repertoire and have benefited many faculty clients since. And along the same lines, the Project Manager's meeting agendas and notes provided valuable insights any time the team needed to reflect back on a nuance of the process in order to effectively iterate design processes. As instructional designers, we should not be strangers to evaluating products and processes, and it is just as important to use data to improve our own products and processes as well.

References

6th Annual LMS Data Update. (2018, October 6). Retrieved from https://edutechnica.com/2018/10/06/6th-annual-lms-data-update/

Adeyinka, T. (2011). Reliability and factor analysis of a blackboard course management system success: A scale development and validation in an educational context. *Journal of Information Technology Education: Research, 10*, 55–80.

Adeyinka, T., & Mutula, S. (2010). A proposed model for evaluating the success of WebCT course content management system. *Computers in Human Behavior, 26*(6), 1795–1805.

Beck, D., Black, E., Dawson, K., DiPietro, M., & Jinks, S. (2007). The other side of the LMS: Considering implementation and use in the adoption of an LMS in online and blended learning environments. *Tech Trends, 51*(2), 35–39.

Berggren, A., Burgos, D., Fontana, J. M., Hinkelman, D., Hung, V., Hursh, A., & Tielemans, G. (2005). Practical and pedagogical issues for teacher adoption of IMS learning design standards in Moodle LMS. *Journal of Interactive Media in Education, 2005*(1).

Coates, H., James, R., & Baldwin, G. (2005). A critical examination of the effects of learning management systems on university teaching and learning. *Tertiary Education and Management, 11*, 19–36.

Dick, W., Carey, L., & Carey, J. O. (2015). *The systematic design of instruction*. New York, NY: Pearson.

Guzdial, M. (2015). Learner-centered design of computing education: Research on computing for everyone. *Synthesis Lectures on Human-Centered Informatics, 8*(6), 1–165. https://doi.org/10.2200/S00684ED1V01Y201511HCI033

Hassan, T., Edmison, B., Cox, L., Louvet, M., Williams, D., & McCrickard, D. S. (2020). Depth of use: An empirical framework to help faculty gauge the relative impact of learning management systems tools. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*. https://doi.org/10.1145/3341525.3387375

Kowch, E. G. (2005). Do we plan the journey or read the compass? An argument for preparing educational technologists to lead organisational change. *British Journal of Educational Technology, 36*, 1067–1070. https://doi.org/10.1111/j.1467-8535.2005.00577.x

Leitner, P., Khalil, M., & Ebner, M. (2017). Learning analytics in higher education: A literature review. In A. Peña-Ayala (Ed.), *Learning analytics: Fundaments, applications, and trends: A view of the current state of the art to enhance e-learning* (pp. 1–23). Cham, Switzerland: Springer International Publishing.

McGill, T. J., & Klobas, J. E. (2009). A task–technology fit view of learning management system impact. *Computers & Education, 52*(2), 496–508.

Mwalumbwe, I., & Mtebe, J. S. (2017). Using learning analytics to predict students' performance in moodle learning management system: A case of Mbeya University of Science and Technology. *The Electronic Journal of Information Systems in Developing Countries, 79*(1), 1–13. https://doi.org/10.1002/j.1681-4835.2017.tb00577

Ozkan, S., & Koseler, R. (2009). Multi-dimensional students' evaluation of e-learning systems in the higher education context: An empirical investigation. *Computers & Education, 53*(4), 1285–1296.

Resier, R. A., & Dempsey, J. V. (2007). *Trends and issues in instructional design and technology* (2nd ed.). New York, NY: Pearson.

Samarawickrema, G., & Stacey, E. (2007). Adopting Web-Based Learning and Teaching: A case study in higher education. *Distance education, 28*(3), 313–333.

Sanders, K., Ahmadzadeh, M., Clear, T., Edwards, S. H., Goldweber, M., Johnson, C., . . . Spacco, J. (2013). The Canterbury QuestionBank: Building a repository of multiple-choice CS1 and CS2 questions. In *Proceedings of the ITiCSE working group reports conference on Innovation and technology in computer science education-working group reports* (pp. 33–52). New York, NY: Association for Computing Machinery. https://doi.org/10.1145/2543882.2543885

West, R. E., Waddoups, G., & Graham, C. R. (2007). Understanding the experiences of instructors as they adopt a course management system. *Educational Technology Research and Development, 55*(1), 1–26.

Wilcox, D., Thall, J., & Griffin, O. (2016). One canvas, two audiences: How faculty and students use a newly adopted learning management system. In *Society for Information Technology & Teacher Education International Conference* (pp. 1163–1168). Association for the Advancement of Computing in Education (AACE).

Daron Williams is the Director of Instructional Design and Development for Technology-enhanced Learning and Online Strategies (TLOS) at Virginia Tech. He comes from a professional background originally in broadcasting and journalism, but gradually wound his way toward technology-enhanced instructional design over the years.

Daron has earned a master's degree in Communication from Virginia Tech and is currently a PhD candidate in Curriculum and Instruction in the School of Education. His research interests include learner motivation and engagement in technology-enhanced courses and in extended reality environments.

Larry Cox II Larry currently works as an Instructional Designer at Virginia Tech and has been working in Instructional Design (ID) for about 5 years. He completed his MA in Instructional Design at Virginia Tech and is currently working on his PhD. Prior to working in ID, he worked in IT as a network and system administrator. Outside of work, you can usually find him gaming, rock climbing, weightlifting, or enjoying a good beer.

Margaret Ellis Margaret currently serves as an Associate Professor of Practice and is focused on instructing and designing curriculum for CS2104 Problem Solving in Computer Science and CS2114 Software Design and Data Structures. She received her BS in Mathematics, then her MS in Computer Science, both from Virginia Tech. Margaret taught Secondary Math and Computer Science in Maryland and Virginia and has industry experience as a System Engineer, Consultant, Trainer, Project Manager, and Developer. She also created her own software business which involved designing and developing educational apps for individuals with special needs, as well as customizing and launching the use of handheld devices within many school systems. Margaret began teaching at Virginia Tech in 2013 and enjoys integrating her various professional experiences within her courses to provide students with real world perspectives and training in contemporary topics and skills for both research and industry. She is interested in Computer Science Education and approaches to support individuals from underrepresented groups in computing.

Bob Edmison is a Collegiate Assistant Professor in the Department of Computer Science at Virginia Tech. His research interests center on developing new ways to provide feedback to computer science students, using tools and techniques originally developed for software engineering. He is also the Coordinator of Online Teaching and Learning in the Department of Computer Science. Previously, Bob was the Director of Software Development for Technology-enhanced Learning and Online Strategies, TLOS, in the Division of Information Technology at Virginia Tech. He has also done extensive work in industry and not-for-profits, as well as research for the U.S. Department of Energy in the areas of materials management, experimental simulation and visualization, and nuclear non-proliferation.

Taha Hassan is a doctoral candidate with the Department of Computer Science at Virginia Tech. He works on issues in human-centered design of information systems, especially online recommendation of learning resources.

M. Aaron Bond Dr. Aaron Bond has worked in the field of instructional technology, distance education, and professional development for more than 20 years. He has served as a corporate trainer, face-to-face classroom instructor, online instructor, and as a secondary principal. Currently he serves as the Senior Director for Learning Services in Technology-enhanced Learning and Online Strategies at Virginia Tech where he oversees faculty professional development, learning technologies, and instructional design.

He has an earned BA in History from Mary Baldwin College, an MA in Interdisciplinary Studies from Fort Hayes State University, an EdS Educational Leadership and Policy Studies from the University of Virginia, and a PhD from Virginia Tech in Curriculum and Instruction.

His research includes authentic learning experiences, communities of practice, socio-cognitive factors on learning, and distance learning. He is the co-author of two books: “Building Virtual Communities of Practice for Distance Educators” and “Systems Thinking for Instructional Designers: Catalyzing Organizational Change.”

Quinn Warnick is the Deputy Executive Director of Technology-enhanced Learning and Online Strategies (TLOS), Virginia Tech’s academic technologies unit. TLOS partners with faculty in all disciplines to design successful digital learning environments and experiences for traditional, online, and hybrid courses. Prior to his current appointment, Dr. Warnick served as an Assistant Professor of Digital Rhetoric in the Department of English and Co-Director of the Center for Applied Technologies in the Humanities. His teaching, research, and administrative work lies at the intersection of digital pedagogy, virtual communities, and emerging technologies.

Virginia Clark holds an MAEd degree in Curriculum and Instruction with a focus on Instructional Design and Technology from Virginia Polytechnic Institute and State University. She currently practices instructional design in southwest Virginia.

Daniel Yaffe Dan currently works as a Learning Technologies Specialist at Virginia Tech and has been working in various positions in Technology-enhanced Learning and Online Strategies, TLOS, in the Division of Information Technology at Virginia Tech for the past 7 years. He completed his Master of Arts in Education, Training, and Instructional Technology at Indiana University of Pennsylvania and is currently working on his PhD at Virginia Tech. He has a professional background in the economic and insurance industry centered on employee development.

Molly Domino is a current PhD candidate at Virginia Tech studying Computer Science Education and seeks to become a teaching professor to help new students gain an understanding and passion for the field. Before graduate school, Molly worked for 5 years as a software developer, first for Cognius LLC and then for the Global Obesity Prevention Center.

Derek Haqq is an educator, computer science researcher, and management professional with extensive teaching experience at tertiary and secondary level. Derek Haqq holds a BSc and MSc in Computer Science, a Postgraduate Certificate in Higher Education, and several professional and graduate certifications in technology and business administration. He is currently a PhD student at Virginia Tech. His research interests include technology on the trail and exploring ways that technology may be designed and leveraged for teaching and learning, relational maintenance, and social recreation.