Paper ID #23617

From Learning to CAD to CADing to Learn: Teaching the Command, Strategic, and Epistemic Dimensions of CAD Software

Dr. Dean Nieusma, Rensselaer Polytechnic Institute

Dean Nieusma is Associate Dean for Curricular Transitions, Associate Professor in Science and Technology Studies, and Director of the Programs in Design and Innovation at Rensselaer.

Dr. James W. Malazita, Rensselaer Polytechnic Institute

James Malazita is an Assistant Professor in Science & Technology Studies and Games & Simulation Arts & Sciences at Rensselaer Polytechnic Institute.

Ms. Lydia Rebekka Krauss, Rensselaer Polytechnic Institute
Miss Andrea M. Ukleja

2016 graduate of Rensselaer Polytechnic Institute with a B.S. in Design, Innovation and Society & Mechanical Engineering. Currently employed by Hasbro, Inc. as an Associate Reliability Engineer.

Timothy Andrews, Rensselaer Polytechnic Institute

Timothy Andrews is an undergraduate student at Rensselaer studying Mechanical Engineering and Programs in Design and Innovation.

From Learning to CAD to CADing to Learn: Teaching the Command, Strategic, and Epistemic Dimensions of CAD Software

Introduction

Computer Aided Design (CAD) instruction, required for most engineering students at Rensselaer Polytechnic Institute, takes a command-centered approach, demanding that students successfully replicate sample engineering models, assemblies, and associated drafts. This educational approach is effective in helping students gain sufficient declarative and procedural knowledge—that is, knowledge about the relationships between user inputs and corresponding software responses in order to replicate pre-specified component parts and assemblies. Drawing on Ivan Chester's framework of CAD knowledge domains, this method succeeds with declarative and procedural knowledge but falls short in teaching strategic knowledge required to effectively employ CAD *as a design tool*.

According to Chester, declarative and procedural knowledge center on command sequences that lead to desired object geometries, whereas strategic knowledge involves higher-order design decision making [1]. For Chester, strategic knowledge is critical because it is the primary determinant of both the efficiency with which a design is completed and the extent to which a design captures "design intent," or the ability of the resulting artifact to achieve requirements as it is iterated over time. In this essay, we argue that learning CAD software also involves "epistemic" dimensions of knowledge—knowledge not about discrete facts and processes, but about the interrelations of CAD and design more broadly, or how the design tool shapes the design process and its outcomes. Epistemic knowledge, then, is less about "what" CAD practitioners know, but rather "how" they know it.

Our research, funded by an internal pedagogical innovation grant, hypothesizes that a more engaging learning process—emphasizing epistemic CAD knowledge and involving student collaboration—could convey epistemic expertise alongside declarative, procedural, and strategic knowledge, thus teaching CAD beyond its "command knowledge" components [2]. To test this hypothesis, a cross-disciplinary team of instructors and undergraduate students—representing engineering, design, social sciences, and humanities—conducted a series of CAD instructional workshops employing a variety of educational exercises and assessing student knowledge acquisition in declarative, procedural, strategic, and epistemic domains. These exercises intersected the teaching of CAD command knowledge with strategic and epistemic knowledge via open-ended design prompts responded to by students in collaborative groups and overseen by a team of student-instructors. We assessed students' CAD learning in conjunction with their responses to the design prompts. Our findings indicate students were, in fact, more engaged by the epistemic emphasis of our educational experiments, and yet continual attention to command knowledge was still required throughout all levels of CAD instruction. Consistent with Chester, these findings suggest that while command-centric learning should remain integral to CAD instruction, emphasizing higher-order design thinking in the context of CAD instruction can change how students relate to the tool—and through this can be effective in transforming student understanding of how CAD inflects engineering design output.

Background and Context

This project was motivated by a set of three distinct but intersecting forces: 1) Demand by our students to add CAD instruction to an established social sciences-centered design and innovation undergraduate programming (the Programs in Design and Innovation, or PDI); 2) Recognition that PDI faculty were weak with CAD tools, and thus there was a need to teach CAD from the perspective of our strengths in human-centered design and design thinking; and 3) A funding opportunity to support faculty who sought to integrate interactive educational technologies into their teaching. The authors pursued the funding opportunity by proposing to use collaboration tools, specifically employing the Barco ClickShare screen-sharing technology and large-screen displays, to develop a set of CAD+Design teaching modules that could eventually be deployed by other programs that teach creativity, CAD, or design at the introductory level. The proposal was approved for funding, and a set of instructional experiments were carried out and the modules were developed and tested. These activities are detailed in this paper.

The demand for CAD instruction within PDI had two origins. Roughly three quarters of our students are engineering dual majors [3], who must take during their first year an introductory 1-credit CAD course that focuses on command knowledge. The course requires students to replicate sample engineering models, assemblies, and associated drafts using Siemens NX. The course is offered in a "flipped" format, with videotaped lectures that must be viewed outside of class time, so scheduled class time is dedicated to instructors and TAs reviewing student work and troubleshooting where needed. This educational approach is effective in helping students gain sufficient command knowledge—that is, knowledge about the relationships between their inputs as software users and the corresponding responses of the software in order to replicate pre-specified component parts and assemblies.[1]

Despite being offered in a flipped format, our students find the course unengaging. Furthermore, they find it difficult to reconcile their creative design skills with their command-centered NX skills. They end up not being able to effectively "work" in NX *during the design process*, because they experience the rigidity of the modeling process to confound their creative design process. They find themselves repeatedly needing to "start from scratch" in their modeling, as prior iterations prove unworkable as they move more deeply into design specification. Also, the effort to build a model according to their CAD instruction primarily required rote "grunt work," neither drawing on nor contributing to their design creativity. The other origin of student demand for CAD instruction in our program came from the remaining quarter of our students who are not engineering dual majors and thus have little or no exposure to any CAD software. These students want to learn CAD in the context of their already-initiated design education.

Being a social sciences-administered design program, our program's faculty are, for the most part, inexpert with CAD. This includes the faculty co-authors, one of whom (Malazita) is a digital designer and humanist who is competent with Autodesk Maya and the other of whom (Nieusma) is a social scientist and design scholar with undergraduate training in engineering and

---

[1] This approach also has the notable benefit of reaching a very large number of engineering students with limited instructional resources. Given the much smaller size of our design program, extensibility of instruction is less of a concern to us.

elementary knowledge of CAD. We wanted to find a way to introduce basic CAD skills that aligned with our established expertise in design process. For institutional and curricular reasons, we have a limited number of credit-hours to work with in the program, and thus cannot dedicate a course exclusively to CAD instruction. Hence, the proposed project entailed the creation of a sequence of educational modules that could be integrated into existing human-centered design courses by instructors without CAD expertise. Our strategy was to integrate CAD with design thinking skills in a way that could be comfortably moderated by instructors with limited CAD expertise and supported by those students competent with the CAD platform used. Instructors could moderate the overall process, while student assistants could convey command knowledge and troubleshoot where needed.

For this project, we chose to use Rhinoceros 3D modeling software (henceforth, simply Rhino), because we assessed it to lie between traditional engineering CAD software (e.g., Siemens NX, Autodesk AutoCAD, and SolidWorks) and modeling software dedicated to computer graphic imagery (CGI) applications (e.g., Autodesk Maya and The Foundry's Modo). We hoped Rhino also would enable even novice students to make relatively quick mockups with sophisticated geometry, and thereby would encourage early design concept exploration.[2]

Extending Chester's Domains of CAD Knowledge: Command, Strategic, and Epistemic

In his 2007 article, "Teaching for CAD Expertise," Ivan Chester makes the case for increased attention to the development of strategic knowledge in CAD instruction [4]. Drawing on wide-ranging research on CAD instruction, spatial visualization, and cognitive science approaches to expertise, Chester provides a model for CAD expertise that includes three primary knowledge components: declarative, procedural, and strategic. The first two, declarative and procedural knowledge, are both in what Chester calls the "command" domain—both are types of "command knowledge" in that they are tied directly to understanding of and facility with the various commands available to a CAD user. *Declarative knowledge* is knowledge of the specific commands available to the user and what those commands do. In many but not all instances, these commands are common across platforms. Chester provides the examples of mirroring lines, copying objects, and creating solids by extrusion. *Procedural knowledge* is understanding the particular steps required to successfully execute a given command within a given system, and this knowledge is often specific to each software package.

Strategic knowledge exists apart from the particular commands available within any given CAD software package and entails what Chester refers to as "metacognitive" processes—processes that involve "planning, monitoring, and revising" [5]. Strategic knowledge is important because it enables operators to work efficiently, completing CAD models more quickly. More importantly, strategic knowledge also enables operators to more effectively capture "design intent," or the ability of the model to achieve design requirements as it is iterated over time. In other words, a high level of strategic knowledge allows the rapid creation of models that can be easily modified as design revisions are made. Chester designs his educational experiment to test

[2] Despite this hope, we found premature concept lock-in a risk for novice CAD-learners across platforms, as they expressed reluctance to iterate their concepts when such iteration risked losing their investment in a model.

whether strategic knowledge can be learned simultaneous to, and without distracting from, the learning of command knowledge. His results are affirmative on both counts, and with a high degree of statistical significance. Chester also tests the impact of a range of other variables on students' learning outcomes, with one notable finding: Students who previously received CAD training using a command-centric approach did poorer in learning strategic knowledge than novices, suggesting that command-centric learning of CAD may actually interfere with students' ability to later acquire strategic knowledge, but not vice versa.

We find Chester's model useful as a way of framing our thinking about the need for distinct types of knowledge in designing objects using CAD software. In particular, the layering of command knowledge (both declarative and procedural) and strategic knowledge makes analytically distinct the differences between learning how to *use* CAD and how to *think through* the use of CAD. However, we argue for a more expansive definition of the kinds of "strategic knowledge" involved in CAD use. Our preference is to frame strategic knowledge in a way that goes beyond merely knowing how to work efficiently and preserving design intent. Borrowing from insights in Science and Technology Studies (STS), we argue that CAD software also operates "epistemically" [6] by contributing to users' understandings of design and the design process. Through its affordances and constraints, the CAD tool itself shapes the user's design process and, hence, its outcomes.

How CAD software shapes a designer's work ranges from the mundane and routine to the systemic and structural. For beginning and intermediate-level CAD learners, the tools/affordances made most apparent by the software interface direct their design decisions, both in terms of function and aesthetics, especially in early-stage design exploration. In informal preliminary testing with our undergraduate students, those who attempted early-stage design "concepting" using SolidWorks and NX tended to produce models with strongly rectilinear forms, whereas students experimenting in Maya, a software package designed for CGI and organic modeling, tended to produce models with curvilinear geometry. This bias could be explained by the students' lack of mastery over the software as well as the students' use of CAD to "free sketch" their design concepts rather than to model concepts whose design specifications were already determined (as is utilized in the NX instructional approach described above). However, even experienced CAD students quickly encountered both real and imagined constraints on the software's ability to realize particular forms of geometry.

More importantly, students with prior experience in different CAD programs imagined the distinctions between rectilinear and "organic" form differently. When asked to choose an "organic shape" to model using engineering-centered CAD software, students who were experienced with that software selected Rensselaer's Experimental Media and Performing Arts Center (EMPAC) building, an otherwise rectilinear building with a curvilinear roofline (see Figure 1). Students and faculty experienced in CGI-oriented modeling software balked at what they perceived as these students' limited understanding of organic form, and suggested instead that they attempt to

Figure 1. Rensselaer's EMPAC Building

model an animal, in this case a sinewy mountain lion. We highlight this anecdote because it illustrates the subtle ways that CAD modeling, and the accompanying educational structures, can profoundly impact how students come to know what CAD is and how it works. In our case, students experienced with different software and who came from different disciplinary backgrounds saw the world differently (e.g., what constituted an "organic shape"), and hence made different analytic "cuts" in the world. Their design decisions, though admittedly mundane in this case, were structured by those analytic cuts [7].

Understanding the potential for CAD instruction to explore *command* (encompassing declarative and procedural), *strategic*, and *epistemic* dimensions opens up new possibilities, especially for the teaching of CAD in interdisciplinary design contexts. Adding the epistemic dimension to Chester's command and strategic dimensions allows students to understand how CAD software shapes their approach to knowing the world as a designer, or how the analytic cuts encouraged by CAD shapes students' design problem and solution framing even *outside* of CAD platforms. In other words, education about the epistemic dimensions of CAD software gives students the skills to see, articulate, and acknowledge that CAD can have an implicit influence, and thus the ability to separate those influences from their own design processes. Here, a social scientific lens on the CAD learning activity can be productively integrated with the more technical approach of teaching command and strategic CAD knowledge.

While all students whose career paths intersect with CAD should gain practical and theoretical expertise with relevant CAD platforms, students whose educational experiences and potential careers may span different disciplinary domains would benefit by being agile in terms of how they talk about what they know. In particular, these practitioners need to be able to translate their practical expertise and disciplinary understandings into other domains: moving from, say, engineering to design, or design to marketing. Understanding CAD as having strategic and epistemic dimensions, along with the technical, can help designers both understand and describe these translations. In addition, being introduced to the command, strategic, and epistemic dimensions of CAD platforms simultaneously may be an important identity-building exercise for participating students, in that it may provide them with a framework to understand their own struggles as they seek to span different disciplinary traditions [8].

Two major hypotheses guided the development and testing of our approach to CAD instruction that integrated command, strategic, and epistemic dimensions. First, extending the logic of Chester's approach, we hypothesized that we could add the epistemic dimension to the command and strategic dimensions without compromising student learning of the latter. While as interdisciplinary social scientists and design faculty we are particularly interested in the epistemic dimensions of design, we also wanted to take care to not under-emphasize the importance of learning the practical affordances of the toolset or their strategic application in design. Second, we hypothesized that integrated instruction would help students better understand the array of material and social factors that influence their design decisions, both consciously and unconsciously, thereby enabling them to better understand and describe their own design process.

Module Development

The authors led an interdisciplinary research team of design, engineering, and communication undergraduate students to develop and test a series of integrated command-strategic-epistemic CAD learning exercises. Due to the nature of PDI, which emphasizes group work and collaborative design pedagogy, the team's student research leads felt it was important to design the CAD experiences as group learning assignments. The instructors supported this suggestion for three reasons: 1) CAD learning as a group exercise aligned with our program's typical teaching and learning approaches; 2) we assumed learning CAD as a group exercise would be more engaging than the flipped-classroom model of CAD instruction that our engineering students were already dissatisfied with; and 3) interdisciplinary group learning would provide students exposure to other disciplinary ways of knowing and ways of designing in CAD and, hence, we assumed, it would facilitate learning of strategic knowledge.

Although the integrated CAD learning experiences were designed specifically for the needs of PDI students, we desired to develop an approach that would be applicable beyond the immediate context of our research team and students. First, the modules had to be manageable by instructors with limited exposure to CAD/Rhino (supported by students who had strong Rhino command knowledge). Second, our internal grant funding stipulated that the learning activities we developed should be transferable across classes and departments. The authors thus decided to construct the CAD+Design activities as a series of formalized learning modules that could be integrated in various disciplinary and classroom settings across our campus.

The set of modules was designed to take about 10-11 hours total to complete (over 6-9 sessions), so that modules could be incorporated into existing classes on campus or run as a series of extracurricular workshops. The initial project design called for the development of five modules, each following the same approach of integrating command, strategic, and epistemic knowledge and each building upon the foundation provided by the prior. Beta testing quickly revealed that students needed a primer prior to our first planned module to become oriented to the generalized approach and establish basic recognition of how design (sketching, concepting, CADing) can be approached in diverse ways leading to distinct, equally acceptable solutions. As a result, a sixth "proto-module" was developed to serve that need and inserted at the front. While Modules 2 through 6 require students to interact directly with CAD software, Module 1 entails multiple iterations of collaborative "rapid sketching" exercises, where students are provided a narrative description of a three-dimensional object and then expected to hand-sketch that object. Because the resulting sketches convey that students imagine the textually-described object in radically different ways, this exercise helps students recognize how much unconscious work is done by designers as they translate physical descriptions and functional specifications into material and geometric forms. Table 1 shows the complete module list.

Table 1: CAD+Design Modules

Module 1	Think Outside the Box	1.5 hours
Module 2	Fail Fast – Command Hunt	1 hour
Module 3	Early-Stage Divergence	2-3 hours
Module 4	Pivoting and Plussing	1 hour
Module 5	User-Centered Design	2 hours
Module 6	Research-Driven Ideation	3 hours

Modules 2 through 6 all center around a unique "design problem" that requires students to learn and experiment with command and strategic-level CAD affordances. These affordances are then "epistemically contextualized" through written reflections, group discussion, and collaborative, real-time design iteration. For example, Module 3, "Early Stage Divergence," features computationally aided brainstorming exercises that require students to use Rhino to rapidly proliferate multiple rough "solutions" to a design prompt. The module introduces early-level Rhino functions (such as lofting, shelling, and Boolean operations) and encourages students to make use of those functions throughout their CAD brainstorming. After modeling two or three rough design solutions, students share and discuss their solutions in groups. Then they are asked to describe to the group their thought process and the command sequences they used for generating each model. Finally, they are asked to iterate their own solutions based upon feedback and insights from their group members. This process is repeated twice during the module. Students are thus exposed to command knowledge (the declarative and procedural knowledge required to work in Rhino), strategic knowledge (that there are multiple ways of modeling solutions to a design problem, both in terms of what is modeled and how it is modeled, and that some solution paths are more efficient than others), and epistemic knowledge (reflecting on how the CAD tool impacted their work and sharing their problem-solving and solution-exploration processes). See the Appendix A for a summary of each module, including the associated learning objectives.

Though the modules were initially conceived of as being applicable across CAD platforms, the research team quickly realized that the context of the CAD software itself had a profound impact on the development of each exercise. In retrospect, this should not have been a surprise: As discussed above, different CAD software packages are designed to fit in at different stages of the design/production process and for different design disciplines. For example, NX works best if users have a well-defined mechanical concept that must first be translated into digital and algorithmic space so that it can later be translated into a manufacturing environment. Maya, in contrast, encourages more exploration and experimental modeling, even when translating pre-rendered 2-dimensional imagery into 3-dimensional space. Maya also presumes that digital models will rarely, if ever, need to be translated into physical media beyond the screen, and therefore exchanges dimensional and physical precision for "looks good" dimensional and physics *simulation*. Different CAD software, in other words, not only presumes different user bases, but works to "configure" their users' actions and knowledge in ways that align with different disciplinary or industry design traditions [9, 10].

Further complicating our choice of CAD platform was the diverse disciplinary and educational backgrounds among the research team members. Before earning a PhD in social sciences, Nieusma received an undergraduate degree in mechanical engineering and worked in the automotive industry. Nieusma was thus more comfortable with engineering-centered CAD programs like SolidWorks than Malazita. Malazita, in contrast, received an undergraduate degree in digital media design and a PhD in communication and cultural studies, and was previously employed in the games industry. Malazita had difficulty adjusting to the affordances and constraints of engineering-centered CAD software, and was much more comfortable in CGI-centered CAD like Maya. Krauss, Ukleja, and Andrews, like other PDI student, were most familiar with NX, but also found the software constraining for the type of design work they needed to do in our interdisciplinary design program.

After much discussion among team members, we ultimately settled on Rhino, an industry-standard CAD package in industrial design and architecture. Rhino offered several features that allowed the team to develop modules that were at once relatively accessible to users across disciplines and relatively translatable into both engineering and CGI design processes. Because Rhino's primary user base conceives of the platform as both a visualization/communication tool and a design-for-manufacturing tool, Rhino allows dimensional design work (though not as extensively as NX or Solidworks), while also enabling exploratory, procedural, and freehand modeling (though not as flexibly as Maya). Rhino also contains plugins, dimensioners, and exporters for translating Rhino files into formats readable by CGI renderers, CNC manufacturing systems, and 3D printers. Rhino is thus doubly helpful, as its wide array of affordances gave the design team latitude to experiment with concept-driven module formats, while its epistemic and functional translatability provide flexibility for modifying the modules to work with different CAD packages in other educational contexts.

Module Assessment

The six modules created by the design team were iterated over time based on student feedback, formal and informal deployment in classrooms and workshops with mid-level Rensselaer students, and informal testing with advanced, college-targeting high school students. Module testing and iteration occurred three times over a year and a half with three different user groups. Phase I testing occurred from Fall 2016 to Spring 2017, during which prototypes of the six modules were tested in two 3-hour extracurricular workshop sessions developed for PDI students. The two workshops accommodated 15 students each, mostly overlapping, and tested two modules (1, 3) and four modules (2, 4, 5, 6), respectively. After incorporating student feedback from Phase I testing, the modules were iterated and deployed in Summer 2017 for Phase II testing. This phase entailed two week-long design immersions for college-targeting high-school students, most of whom had no prior CAD experience. The design immersion programming included introductions to human-centered design and 3D printing in addition to the CAD+Design modules. This phase of testing was mostly due to convenience, and quantitative data was not gathered. Phase III testing occurred in the context in one of the PDI studio courses, taught by Nieusma and assisted by Ukleja (Fall 2016) and Krauss (Fall 2017). These modules were incorporated into the design of the course, and were delivered over 4 weeks.

As described above, for Phase I testing, students learned Rhino in a collaborative setting enabled by use of the Barco ClickShare screen-sharing technology. This technology allows up to four students to simultaneously project their laptop screens, a feature that enabled students to see how others were responding to each design prompt within the modules. Figures 2 and 3 show students testing one of the modules using the Barco system.

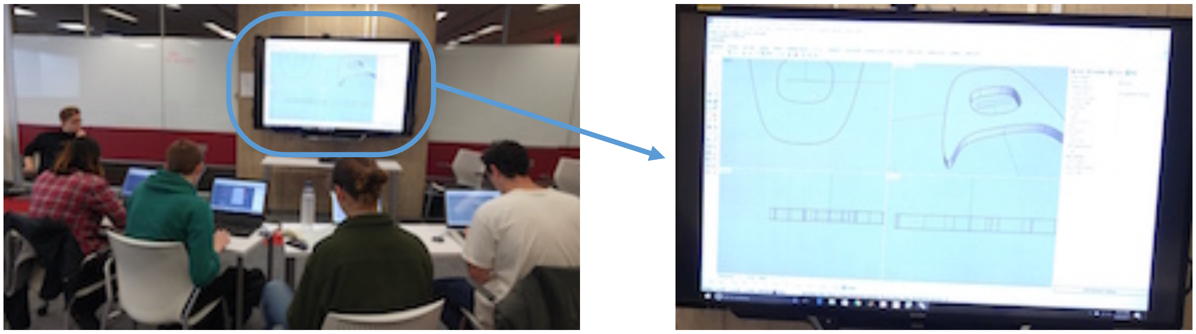

Figure 2: Module testing using Barco screen-sharing technology. Detail: Projection of one student's screen.

Figure 3: Module testing using Barco screen-sharing technology. Detail: Projection of four students' screens.

For each phase of module testing, we collected a range of data from participants. Most of our data was qualitative, relying on our team of student researchers to observe and document challenges faced by the student participants as well as the student module leads (including Krauss, Ukleja, and Andrews). We also collected limited quantitative data for each of the testing phases:[3]

- Phase I: Student comparative self-assessments of their learning experiences and the amount of learning between the NX course format and the Rhino module format. Additionally, student Rhino confidence-level change post-modules.
- Phase II: High-school student CAD learning of command, strategic, and epistemic knowledge compared pre- and post-module integration.
- Phase III: Student Rhino confidence level pre- and post-modules.

Phase I module testing was designed to assess student experience with the first prototype of modules 1 and 3. The primary aim was to user-test the basic logic and approach of the modules in integrating command, strategic, and epistemic dimensions of CAD in the context of open-ended design. Our data collection included three sets of quantitative data, all based on student self-assessment: 1) Comparative perceived effectiveness of the two underlying teaching formats; 2) Comparative overall learning; and 3) Overall comfort level with Rhino post-modules.

---

[3] Different student research teams across our testing phases collected different data sets over the duration of project to-date. Due to loss of data (i.e., the completed CAD files), we were only able to assess the *content* of student CAD work during the summer design workshops. We plan to create a consistent data collection protocol for students doing future module testing, where the entire set of modules is to be formally integrated into a newly created design studio course.

1. Comparative Perceived Effectiveness

The first data set captures how students compared the educational effectiveness of the NX flipped classroom and the Rhino collaborative workshop. Figure 4 shows the results of this comparison, where students experienced the Rhino format to be more effective overall, but with important qualifications. Most notably, student experience with the NX format was bifurcated, with most students disliking it, but a handful strongly liking it. This finding is consistent with the fact that students have a variety of preferred learning styles. Student experience with the Rhino modules showed the opposite profile, with data concentrated in the middle rather than the extremes.

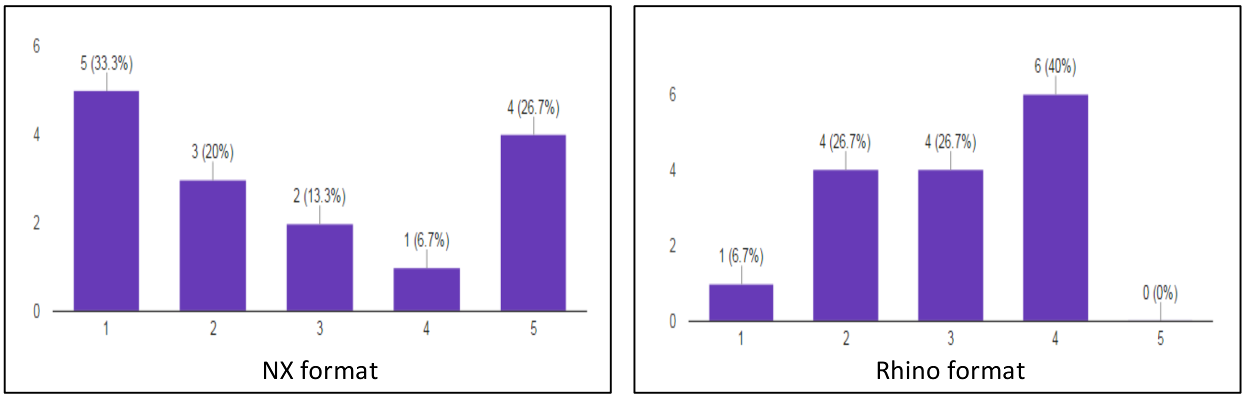

Figure 4. Phase I: Effectiveness of learning format. X axis = Likert low to high; Y axis = responses; N = 15.

2. Comparative Overall Learning

Figure 5 shows the results of student assessments of their overall learning under each course format. Total learning under the NX format parallels student assessment of course-format effectiveness from Figure 4. Total learning under the Rhino module format again concentrated toward the middle, but less consistently than with assessment of course-format effectiveness. Here, we should also qualify the short time duration of the Rhino compared with NX lessons.

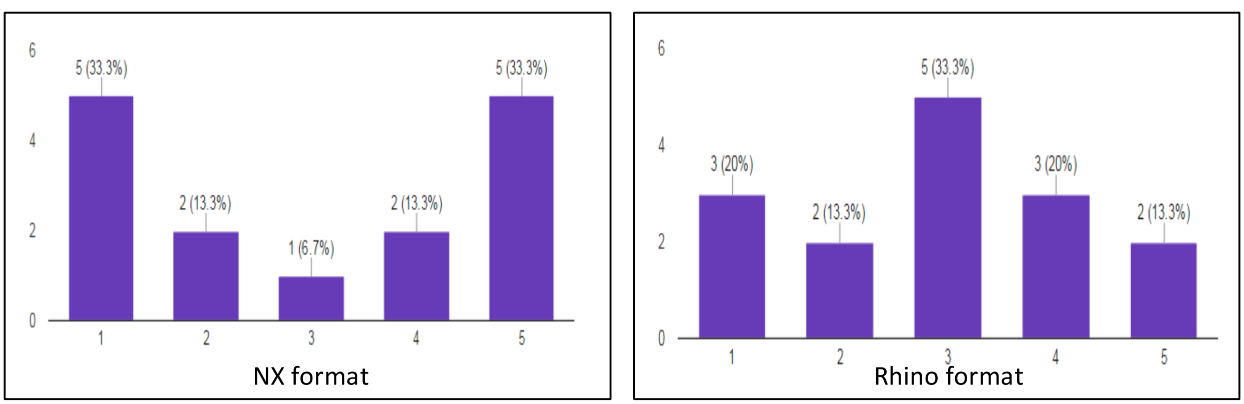

Figure 5. Phase I: Total learning for each format: X axis = Likert low to high; Y axis = responses; N = 15.

3. Overall Rhino Comfort Level

Figure 6 shows student self-assessments of their overall comfort-level with Rhino as a result of participating in the module workshop. Here we see a lukewarm response, including equal numbers of students experiencing no change with those experiencing increased comfort. We also see one student whose confidence level actually decreased as a result of the workshop (possibly due to an iceberg effect, which is discussed in the Findings section below).

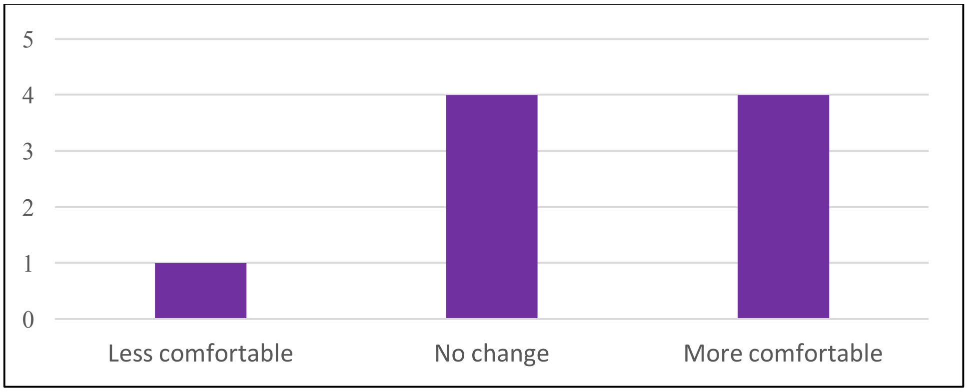

Figure 6. Phase I: Change in Rhino Comfort Level after Training in Modules 5 and 6; N = 9.

Phase II testing was carried out for college-targeting high-school students who participated in a one-week-long summer design immersion workshop hosted by Rensselaer. These students ranged in age from 15 to 18 and worked on open-ended design projects in groups of 3 to 4 members. Group CAD output was analyzed according to a rubric designed by Krauss that assessed command, strategic, and epistemic CAD knowledge as demonstrated by each team's finalized CAD models. (See Appendix B for the full rubric.) The Summer 2016 session included 8 total teams and was taught without use of the modules, whereas the Summer 2017 session included 9 total teams and was taught using a second iteration of the set of modules. Figure 7 shows the results of this round of module testing. Notably no aspect of the groups' work rated above an intermediate level of CAD knowledge demonstration in any dimension (rated 3 on a 5-point scale). Again, it is important to keep in mind the short duration of the lessons. Still, while demonstration of command and strategic knowledge remained relatively constant between the pre- and post-module cohorts, the post-module cohort ranked considerably higher in epistemic knowledge—that is, the post-module cohort was notably better at exploring and experimenting with CAD in order to manifest human-centered design concepts, rather than allowing CAD commands to drive their design decisions.

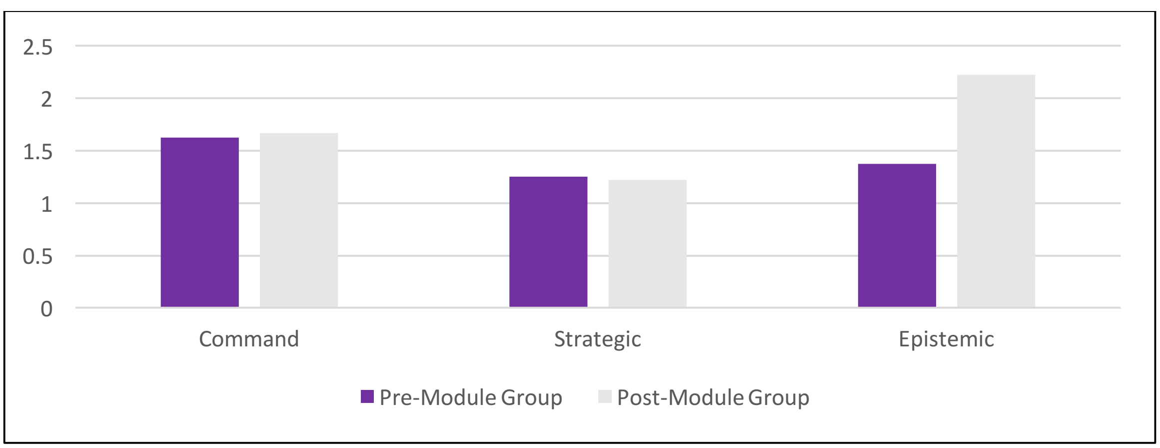

Figure 7. Phase II: Student Learning of CAD Knowledge. Y axis = beginner (1) to expert (5)

As we previously determined the overall effectiveness of the modules in terms of student experience with the learning format, Phase III testing focused exclusively on students' degree of comfort with Rhino as a result of the modules. Figure 8 shows the results of student self-assessment of their comfort level change due to the modules workshop. We see considerable improvement compared to Phase III testing. In addition to having zero students decrease in comfort level, it is worth noting that the four students who experienced no change in comfort level all entered the workshop with intermediate-to-advanced Rhino skills. The introductory nature of modules seems to have aligned poorly with this group's learning needs. Two qualifications are worth making for this test group. First is that the activity occurred during class time. Although participation was entirely optional, and Rhino instruction took place during "open studio" time within the course time block, the vast majority of the students who were enrolled in the course participated (22 out of 26). Second, the vastly improved outcomes relative to Phase I testing was likely due, at least in part, to improvements made in the module design through two rounds of iteration, as well as to Ukleja's and Krauss's increasing experience with the teaching of Rhino in the context of the modules.

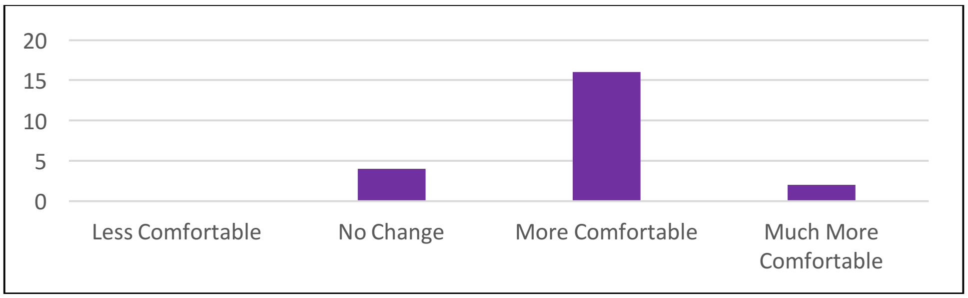

Figure 8. Phase III: Change in Rhino Comfort Level after Training in All Modules; N = 22.

Figure 9 shows student-reported confidence in their Rhino skills prior to and after the training modules. This data is the most concrete evidence of improved student-experience with the Rhino module learning format in terms of students' overall confidence with their modeling skills. Again, there is limited movement of students into the advanced categories, but the large majority of students without Rhino experience gained confidence from modest (1 step improvement) to intermediate (2 step improvement) degrees.

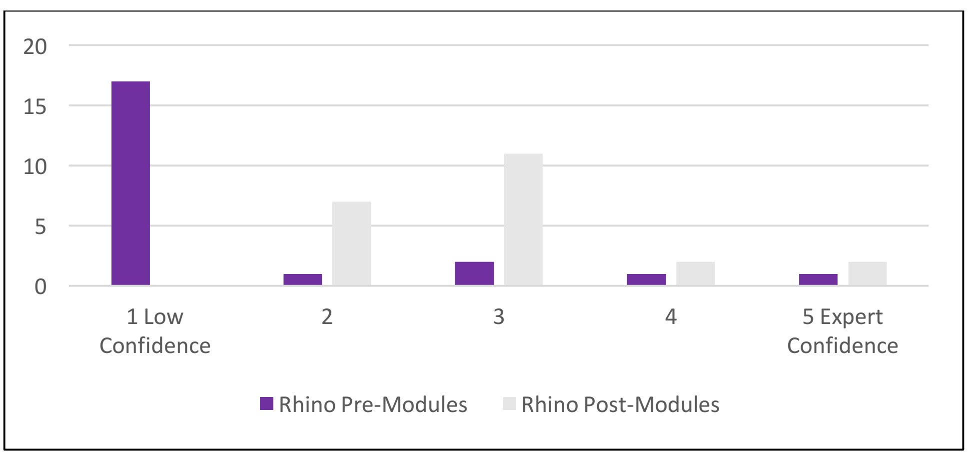

Figure 9. Phase III: Rhino Confidence, Pre- and Post-Module Training; N = 22.

Discussion: Interpreting Quantitative and Qualitative Findings

Our module analysis led to three distinct categories of findings: 1) Students experienced that learning CAD collaboratively and through open-ended design questions was more engaging than the flipped classroom format that most of them were familiar with; 2) Students successfully learned command and strategic knowledge required for basic CAD modeling, but the exploratory nature of the modules also made obvious how limiting basic modeling skills are when trying to create complex objects. This may have led to an "iceberg" effect among students, whereby student learning of entry-level CAD skills enabled them to realize how much more there was to learn, thereby creating a greater sense of anxiety about their CAD expertise than they had prior to the modules; and 3) Students successfully engaged with the epistemic dimensions of CAD, and were able to reflect on how Rhino impacted their understandings of the design process. However, while this was judged as a successful learning outcome by the authors, a notable minority of our students experienced unease with the newly acquired epistemic knowledge of CAD, preferring instead to think of CAD simply as a neutral tool. This appeared to lead to dissonance for some students, who acknowledged the epistemic influence of CAD, but simultaneously framed CAD as "just a tool" in order to feel empowered in their design work.

Students in PDI are used to working in collaborative group environments, and so it was not surprising to us that they had negative experiences with the flipped classroom model of Rensselaer's traditional CAD instruction. In general, PDI students—particularly the majority who are dual-majors with engineering—join our program specifically because they are looking for collaborative, open-ended educational experiences in which to apply their budding technical expertise. What was surprising to us was that students experienced tensions between the collaborative learning environment created with the modules and the traditional CAD lessons. Students reported high levels of engagement in collaborative exercises and appreciated the open-ended design prompts while learning CAD, but they also experienced CAD learning via the modules as inefficient. It seemed as though some students continued to prefer the "technical" learning outcomes of command-centered CAD education even as they expressed appreciation for the open-ended and collaborative aspects of our integrated modules.

This tension between the desire for reflective, epistemic-oriented pedagogy and the desire for "pure" technical knowledge [11] can be partially explained by our students' experience of the constant tension between their engineering and social-science/design identities. However, our second finding—that students gained expertise, but also felt overwhelmed by how much they *didn't* know—may also have contributed to this tension. One of the potential strengths of command-centric CAD pedagogy may be that it instills a sense of technical mastery in students. In command-centric pedagogy, students' perceptions of the depth of functionality of CAD software is greatly limited; they are introduced only to a small set of tools at a time, and are given a sense of agency through how much they are able to accomplish with that limited purview of the software. Our epistemic-oriented CAD pedagogy begins by foregrounding the breadth, depth, and number of options and strategies that are possible within CAD software. Even though our modules quickly erected boundaries to focus their attention, students recognized those boundaries *as artificial*, because they were already exposed to the breadth. They did not see the boundaries as a novice might—as limits of the software that are progressively expanded as they learn additional commands. This is the root of the iceberg effect described above. Such a pedagogical outcome is a double-edged sword. As social scientists, one of our core pedagogical goals is to have students recognize and confront gaps in their knowledge that they never knew existed. However, our students appear to have less confidence in their expertise with CAD than students who actually knew less about the software.

This tension between growing technical competence and diminishing confidence was seen across all three dimensions of CAD expertise. While the first two findings highlighted the tensions in command and strategic knowledge dimensions, our final major finding concerns the epistemic dimension. Post-test discussions revealed that students did, in fact, begin to recognize the diverse ways that CAD is positioned across disciplinary design practices. Crucially for us, they were also able to reflect upon their own implicit assumptions of the role of CAD in their design practice, and began to experiment with more robust and diverse deployments of CAD in their design coursework. However, some students expressed feelings close to resentment about the CAD modules and how they complicated students' understandings of design. It became clear to us that, for a notable minority of students, CAD expertise was seen as a potential cure-all for their self-perceived deficiencies as designers. Students who had difficulty in imagining or expressing interesting design solutions would point to a lack of CAD expertise as the cause, the conceptual block: They could not design well because their instructors had not provided them the technical tools needed to design well. Once they were confronted with the realities of learning CAD—i.e., that CAD influenced them in unconscious ways, that CAD modeling does not replace the need for creative design ability, that their instructors could not possibly cover all of the functionalities of any CAD package—these students lamented what they perceived as the inefficiency of the CAD modules.

These lamentations, however, were valuable findings in their own right, because they helped the research team understand how PDI students were positioning CAD within their own design education. While our students are trained to recognize the practical and conceptual limitations of separating technical expertise from social context, they were still predisposed to framing perceived gaps in their design education as technical-skills deficiencies, just as they were predisposed to framing perceived gaps in their engineering education as social/creative deficiencies.

Implications and Conclusions

The major implications of our work at this stage involve the role of CAD software and instructional techniques to serve as a site of interdisciplinary collaboration and negotiation. As elaborated above, negotiation among the diversely trained co-authors created numerous opportunities for thinking differently about a range of issues: Which CAD software is best for our purposes and why? What, in fact, are our (multiple converging and diverging) purposes regarding CAD software, and do they conflict? How do we understand the relationships among CAD generally, any particular CAD platform specifically, and design? And, most pressingly, how can CAD serve as a tool to mediate our different disciplinary approaches to design from engineering, the social sciences, and the humanities? Certainly, we are not in a position to answer any of these questions definitively, but we have found that collaborating around the challenge of CAD instructional design has forced each of us to reconsider the ways in which CAD works as a design tool, and the ways in which the design process and design outcomes are shaped by the affordances and constraints of any given platform. While the authors are all experienced with the joys and frustrations of interdisciplinary collaboration, situating something as specific and mundane as a CAD software platform at the center of that collaboration resulted in a tremendous amount and array of specific challenges and insights derived directly from disciplinary assumptions about how and where design and innovation occurred.

Extending from the issue of interdisciplinary collaboration is the particular tension between so-called "hard" and "soft" approaches to instruction and instructional analysis and their respective preferences for quantitative and qualitative research. While Nieusma and Malazita identify as qualitative researchers (despite our distinct "technical" backgrounds), and Ukleja, Krauss, and Andrews have fundamental training on both sides, we recognize that most CAD instructors doing engineering education research will lean toward the hard. From this perspective, our research is likely to be interpreted as soft. From the perspective of our humanities and social sciences colleagues, however, our educational research in the domain of CAD instruction places us in patently *technical* space. This apparent contradiction appeals to us, because we see CAD instructional research of this sort as a method of "de-bifurcating" or "decompartmentalizing" the hard and soft approaches to education and the traditional disciplinary alignments along both sides of the hard/soft divide [12]. We also believe that complicating the divide provides a compelling model for interdisciplinary inquiry that can lead to new types of insight [13, 14, 15].

Finally, we see promise in this approach for rethinking many courses focusing on narrow, pragmatic skills acquisition. Again following Chester, we prefer in principle educational models that teach different types of skills and knowledges simultaneously and in an integrated way, just like we experience problems in real life. While we recognize the need for progressive instruction—for using scaffolding in educational programs, for the importance of introducing fundamentals before advanced analytic methods—we do not assume this requires breaking apart learning into its smallest discrete units. Chester's approach invites us to experiment with integrating different sorts of learning goals to see not only what can be combined, but also how the various combinations lead to surprising outcomes. We found we were able to effectively integrate command, strategic, and epistemic dimensions of CAD knowledge in a coherent way, but that our integrated approach was both less efficient than teaching command knowledge alone and not suited to every learning style, even among our interdisciplinary design students.

Now that our modules have been iterated and roughly proven as effective, our next steps for this project will be to carry out systematic module assessment that integrates both student self-assessments and researcher-evaluations of CAD learning across all three dimensions.

Acknowledgements

Research for this paper was supported by a pedagogical innovation grant from Rensselaer's Teaching and Learning Collaboratory. The authors wish to thank the broader set of students who participated in the research activities that led to this paper: Fariha Ahmed, Damiel Faxon, Chris Hoskins, Sebastian Garcia, Katie Kanai, Emily Pearson, Korryn Resetar, Maria Salmon, Sarah Schattschneider, Alex Zylka. We also wish to thank the dozens of students who participated in module testing. This project intersects with related research supported by the National Endowment for the Humanities Office of Digital Humanities, Grant #HD-248450-16.

Appendix A: Modules Summary

<u>Module #1: Think Outside the Box</u>

Description: Students will be prompted to work first individually, and then in teams, to model a box with a hole through its center using multiple unique modeling procedures.

Learning Outcomes:

- Students will become familiar with the basic functionality and interface of the CAD software
- Students will practice creative problem solving individually and as a team
- Students will learn concepts of robustness and design intent with respect to 3D modeling

<u>Module #2: Failing Fast – Command Hunt</u>

Description: Students will have limited commands and time to complete a design prompt. Each time a student "fails," he/she is given another command they are able to use to complete the design.

Learning Outcomes:

- Students will learn that failure is a stepping stone to success, [an important] part of the design process
- Students will learn to embrace failure, not avoid it: "fail fast to succeed sooner"

<u>Module #3: Early-Stage Divergence</u>

Description: Students will go through a series of design "sessions," diverging and converging their thoughts/ideas: "Kill your darlings."

Learning Outcomes:

- Students will learn to continuously expand their thinking throughout the design process
- Students will learn to think outside the box and not restrict themselves when they are brainstorming

<u>Module #4: Pivoting and Plussing</u>

Description: Students will learn how to pivot their ideas and add to/augment them based on newly given/discovered design constraints

Learning Outcomes:

- Students will learn to "kill their darlings"
- Students will learn how to "design through" road blocks and overcome obstacles in their designs

<u>Module #5: User-Centered Design</u>

Description: Students will learn how to collaborate with users to understand needs and transform these needs into design constraints/design decisions.

Learning Outcomes:

- Students will learn interpersonal skills and how to gain user requirements directly from users
- Students will learn how to translate user requirements into design specifications/features/attributes

<u>Module #6: Research-Driven Ideation</u>

Description: Students will re-design previous product to be _______ (give additional constraint that requires research into user group), while retaining the representation of the group members in some way. Students must justify design choices using research findings.

Learning Outcomes:

- Students will solidify the learning outcomes of Modules 1-5
- Students will learn how to "design through" road blocks and overcome obstacles in their designs
- Students will learn how to translate user requirements into design specifications/features/attributes
- Students will practice effectively communicating their designs and design choices to a larger audience

Appendix B: Rubric for Assessing Student CAD Output across Knowledge Dimensions

Ranking	Command Knowledge	Strategic Knowledge	Epistemic Knowledge
1	Simple solids created with basic commands	No or basic interaction among solids created	Limited attention to user experience; poor application for CAD for 3D printing
2	Rectilinear solids created using basic commands	Use of basic Boolean operations	Attention to user experience; poor application for CAD for 3D printing
3	Solids with complex curvature created by extrusion and rotation	Use of Boolean operations with attention to transitions between solids	Basic understanding of affordances and constraints of CAD software
4	Solids with complex curvature created and manipulated	Sub-solids created using original solid	Advanced understanding of affordances and constraints of CAD software
5	Complex, multidimensional solids formed and shaped	Solids created with intention of interaction	Model designed in anticipation of affordances and constraints of CAD software

References

[1] Chester, Ivan. 2007. "Teaching for CAD Expertise." *International Journal of Technology and Design Education* 17(1):23-35.

[2] Chester. 2007.

[3] Nieusma, Dean. 2015. "Analyzing Contexts by Design: A Curriculum and Pedagogy for Integrating Social and Technical Analysis." In *International Perspectives on Engineering Education: Engineering Education and Practice in Context. Vol. I.* Steen Hyldgaard Christensen, Christelle Didier, Andrew Jamison, Martin Meganck, Carl Mitcham, and Byron Newberry, editors. Springer Science + Business Media B.V. Pp. 415-434.

[4] Chester. 2007.

[5] Chester. 2007. P. 27.

[6] Knorr Cetina, Karin. 2009. *Epistemic Cultures: How the Sciences Make Knowledge*. Harvard University Press.

[7] Bucciarelli, Louis L. 1994. *Designing Engineers*. MIT Press.

[8] Nieusma, Dean, and James W. Malazita. 2016. "'Making' a Bridge: Critical Making as Synthesized Engineering/Humanistic Inquiry." *Proceedings of the 2016 American Society for Engineering Education*, ASEE.

[9] Grint, Keith, and Steve Woolgar. 1997. *The Machine at Work: Technology, Work and Organization*. Polity Press.

[10] Downey, Gary L. 1998. *The Machine in Me: An Anthropologist Sits among Computer Engineers*. Routledge.

[11] Latour, Bruno. 2012. *We Have Never Been Modern*. Harvard University Press.

[12] Malazita, James W. Forthcoming. "Translating Critical Design: Agonism in Engineering Education." *Design Issues*.

[13] Malazita, James W. Forthcoming. "Epistemic Infrastructures, the Instrumental Turn, and the Digital Humanities." In *Debates in the Digital Humanities: Institutions, Infrastructures at the Interstices*. McGrail, Anne, Angel David Nieves, and Siobhan Senier, editors. University of Minnesota Press.

[14] Malazita, James W., Dean Nieusma, and Dominic Gelfuso. 2016. "Contextualizing 3D Printing's Contributions to Techno-Creative Literacies." *Proceedings of the 2016 American Society for Engineering Education*, ASEE.

[15] Nieusma and Malazita. 2016.