

Maestro: Designing a System for Real-Time Orchestration of 3D Modeling Workshops

Volodymyr Dziubak^{*†}, Ben Lafreniere^{*}, Tovi Grossman^{*‡}, Andrea Bunt[†], George Fitzmaurice^{*}

^{*}Autodesk Research
Toronto, ON, Canada

{*firstname.lastname*}@autodesk.com

[†]University of Manitoba
Winnipeg, MB, Canada

{*vdziubak, bunt*}@cs.umanitoba.ca

[‡]University of Toronto
Toronto, ON, Canada

tovi@dgp.toronto.edu

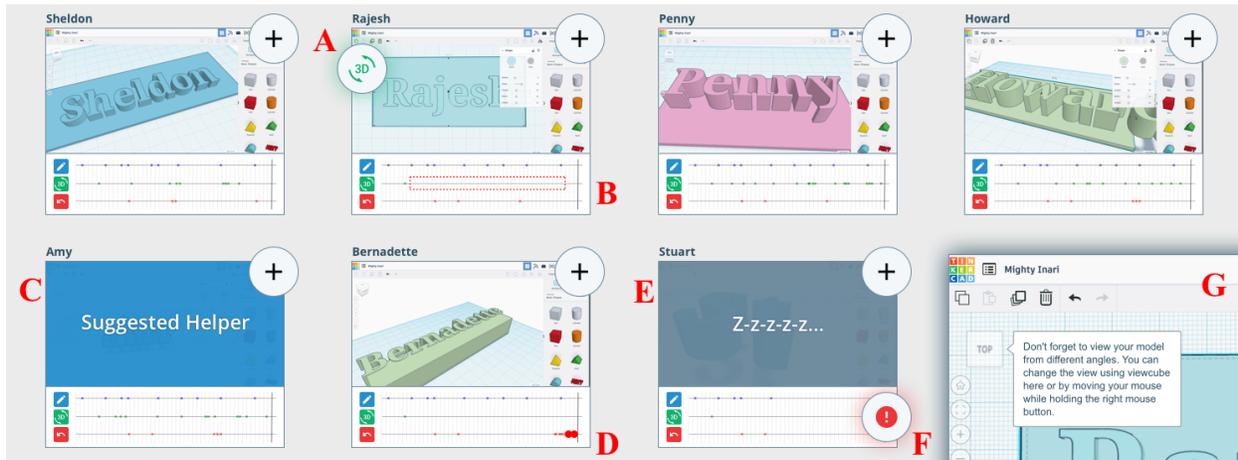


Figure 1. Overview of Maestro's interface. (A) a recommended 3D navigation intervention; (B) system highlights extended periods without 3D navigation; (C) a peer helper recommendation; (D) system highlights multiple consecutive undo/erase actions; (E) inactive student indicator; (F) a recommended "attention" intervention; (G) 3D navigation reminder on student's screen.

ABSTRACT

Instructors of 3D design workshops for children face many challenges, including maintaining awareness of students' progress, helping students who need additional attention, and creating a fun experience while still achieving learning goals. To help address these challenges, we developed Maestro, a workshop orchestration system that visualizes students' progress, automatically detects and draws attention to common challenges faced by students, and provides mechanisms to address common student challenges as they occur. We present the design of Maestro, and the results of a case-study evaluation with an experienced facilitator and 13 children. The facilitator appreciated Maestro's real-time indications of which students were successfully following her tutorial demonstration, and recognized the system's potential to "extend her reach" while helping struggling students. Participant interaction data from the study provided support for our follow-along detection algorithm, and the capability to remind students to use 3D navigation.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the authors must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

UIST '18, October 14–17, 2018, Berlin, Germany

© 2018 Copyright is held by the authors. Publication rights licensed to ACM.

ACM ISBN 978-1-4503-5948-1/18/10...\$15.00

<https://doi.org/10.1145/3242587.3242606>

Author Keywords

Software learning; real-time teaching assistance; analytics.

ACM Classification Keywords

H.5.m. Information interfaces and presentation: Misc.

INTRODUCTION

A common way to introduce 3D design software to children is through in-person group workshops [22]. These workshops are often held in libraries or makerspaces, and typically have a follow-along structure – a workshop facilitator demonstrates how to create an example 3D model, while participants follow along at individual computers. While this approach is generally effective, it presents challenges for both the facilitator and the children participating. Children learn at different paces, leading some to become bored or distracted [17,31], while others fall behind, or become waylaid by usability issues with the software [12,22,31,34]. For the facilitator, it can be a challenge to maintain awareness of the students' progress, so they can set a pace that keeps most students interested and on track, while also taking time to assist students who need extra attention.

In this paper, we investigate the idea of data-driven assistance for workshop orchestration – using real-time analysis of log data from students' individual instances of the software being taught to summarize each student's progress, draw the facilitator's attention to potential issues, and recommend interventions that help students learn the content.

We implement this idea in *Maestro*, a workshop orchestration system for Tinkercad 3D design workshops. Maestro

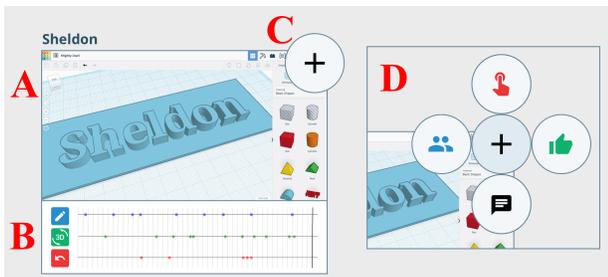


Figure 2. A student widget. (A) live view of the student's Tinkercad window; (B) timelines of command activity; (C) the interventions menu; (D) the interventions menu (open)

presents the workshop facilitator with a dashboard interface containing a widget for each student (Figure 1). Each widget includes an overview of the student's recent activity via an interactive timeline (Figure 2B), and a live view of the student's application window (Figure 2A), to give the instructor a quick sense of the class's overall progress.

Unlike existing classroom awareness tools (e.g. SoftLink¹, NetSupport School², GroupScribbles [27]), Maestro automatically identifies and draws the facilitator's attention to patterns of software activity across all students. For example, during follow-along tutorials, the system detects which students have completed a demonstrated instruction, and prominently highlights this information on the dashboard. Maestro also identifies common challenges faced by individual students, and brings them to the instructor's attention. For example, novices learning 3D design software often forget to look at a model from different 3D viewpoints [22]. Maestro detects this behavior and notifies the instructor with an icon displayed on the student's widget (Figure 1A). If the facilitator chooses, they can act on this notification by sending a quick reminder to the student's screen (Figure 1G).

We designed Maestro based on existing software learning research and refined its functionality by consulting with two workshop facilitators from local makerspaces. To gain preliminary insights into the utility of the system, and the effectiveness of our "follow-along" detection heuristic, we organized and ran an introductory Tinkercad workshop with 13 children, ages 10-17, and an experienced workshop facilitator. The results of this case study indicate enthusiasm for Maestro's features, and suggests that data-driven assistance can help instructors to dynamically adjust the pace of follow-along workshops, can assist them with managing how to split their attention between students, and can provide potentially valuable new perspectives on students' performance.

Overall, this work contributes the design, implementation, and a case study of a system that supports a novel software learning context: group workshops for children learning 3D modeling software. Through a user-centered design process with experienced workshop facilitators, and review of prior literature on children learning 3D modeling software, we identify key design considerations, and illustrate how these considerations can be addressed in a tool that augments in-person instruction for these workshops.

RELATED WORK

This project is related to work on classroom management tools; software learning systems, particularly those that use application log data; and learner analytics systems.

Classroom management tools

A number of commercial classroom management tools (e.g., Softlink and NetSupport School) capture students' screens and broadcast them to the instructor, providing awareness of each student's current state. They also provide the instructor with coarse intervention options, such as freezing a student's input, or taking control over their computer.

Several research systems have also been developed to support general coordination among students and instructors. For example, GroupScribbles [27] extends the concept of sticky notes to digital classroom media, and FireFlies2 supports cognitive offloading through tangible pixel devices distributed through the classroom [36].

Unlike the tools discussed above, which are designed to provide general support for classroom coordination, our system provides contextual information about students' activity in a specific software application being taught. The idea is to help instructors with early detection of potential problems, and to inform targeted learning interventions. Our system also explicitly supports follow-along tutorials for demonstrating how to use software, a common practice in workshops that is not explicitly supported by existing tools.

Also relevant are tools designed to enable remote assistance for learners in Massive Open Online Courses (MOOCs). Codeopticon [20] enables one tutor to monitor and chat with many remote students working on programming exercises. The tutor sees a dashboard showing each learner's code editor, and the system assists them in assessing a student's progress with real-time text diff visualizations and automatic highlighting of compilation errors.

Our system is inspired and informed by Codeopticon's overall approach, but looks at a different context (in-person group workshops) and domain (3D modeling). This changes the problem in several important ways: (1) error states are more challenging to identify because there is no clear corresponding concept of a compilation error; (2) the primary goal is to support in-person instruction of a group (i.e., conveying knowledge to many students simultaneously) rather than multiplexed tutoring (i.e., detecting and correcting the misconceptions of individual students); (3) the system plays a supporting role, rather than mediating all communication between mentor and student; and (4) the attentional demands on the instructor are greater, as he/she must balance instructing students, managing the class, and attending to the system.

Educational data mining and learning analytics

For over 40 years, work in the intelligent tutoring systems community has examined how to create models of students and knowledge, with the goal of automatically personalizing educational content [2,8]. Recent work in this space has sought to extend these ideas, taking advantage of the big data

¹ <http://www.acs-linksystems.com/products/softlink.cfm>

² <http://www.netsupportschool.com/index.asp>

produced by MOOCs [7,29], or enabling *learning analytics* tools, which focus on summarizing and reporting student data to stakeholders, rather than automatic interventions [5]. For example, *open learner models* present students with a representation of their knowledge and progress [19,23,30]. Other systems report on student progress to instructors [3], parents [10], or school leadership [37]. A motivation behind open learner modeling, and learning analytics more generally, is that automated interventions can be brittle, because it is difficult to capture and synthesize all relevant data into a model of the student. Reporting data-driven insights to stakeholders allows them to integrate additional knowledge and expertise, and make the final determination of how to act.

We are inspired and informed by the learning analytics approach. While work in this emerging area has mostly looked at informing instructors about students on timelines of months or years, we are interested in how this approach can be applied in real-time to support group workshops.

Software learning systems

Finally, a number of software learning systems have used data from software logs to enhance software tutorials [24,32], or provide improved help or capabilities within feature-rich software [11,16,18]. These projects, however, focus on an individual learning or using the software on their own. We are interested in how real-time log data from a group of students can be used to enhance workshop instruction and learning experiences. Moreover, we are interested in exploring systems that support software learning in group settings, a topic that has only received limited attention [26].

MOTIVATION AND DESIGN GOALS

Our approach originates in a careful consideration of the challenges faced by children learning 3D modeling software, and the facilitators of children’s 3D modeling workshops. In this section, we review prior research to develop a grounded set of design goals for a data-driven approach to support the facilitators of these workshops.

Past work has shown that children can become distracted during group workshops, or caught up in experimenting with application features, rather than proceeding with the material being taught [9,17,31]. Children often also encounter usability issues with the software, leading to frustration or boredom [12,15,22,31,34]. In their attempts to overcome difficulties, children can experiment with unrelated UI elements in the application, further compounding an initial problem [22,34]. These findings highlight the importance of a facilitator being able to diagnose and address challenges when they first occur, before they become a more serious disruption.

For the domain of 3D design software, past work has shown that children often struggle with understanding the concept of three-dimensional space, and the necessity of viewing a model from different angles [22,34]. This can lead them to work on models from a fixed perspective, producing results that appear correct, but contain serious problems, such as objects floating in space when they should be connected.

Children are also very social in workshops – they often ask and provide help to each other, and they like to compare their work with their peers, and show off their discoveries [12,22,33,34]. Thus, there may be an opportunity to channel some of this social activity into enabling peers to help one another, which is generally believed to be a beneficial practice [35].

For workshop facilitators, the main challenge is keeping aware of the activities in the classroom, and helping participants to overcome the challenges discussed above.

Based on the above points, we developed the following set of design recommendations for software-oriented workshop orchestration systems:

Glanceable awareness of student progress. Our first goal is to provide the facilitator with “at a glance” awareness of students’ progress, both for the class as a whole, and for individual students. This is a challenge for in-person follow-along tutorials, because the facilitator typically cannot see students’ screens, and may even be blocked from seeing the students themselves (e.g., by computer monitors). Moreover, even if they could see the screens for all participants, there is an opportunity for the system to provide semantic awareness indicators of the most important aspects of student progress, to minimize demand on the instructor’s attention.

Automatically identify relevant patterns. To avoid overwhelming the facilitator, the system should take on some of the task of making sense of the incoming data, and present the instructor with higher-level patterns and actionable insights to support their teaching activities. This could take the form of global patterns extending across all students, or synthesizing data from both the instructor (e.g., as they demonstrate operations in the software) and the students. It may also involve identifying important patterns in individual student data, such as common usability problems.

Enable quick interventions that supplement the instructor’s in-person capabilities. For patterns in student behavior that have an obvious response, the system should help automate providing that form of help. For example, to respond to a student who has not used 3D navigation in a while, it may be enough to provide a quick reminder that this capability exists, and that this is a best practice to follow when doing 3D modeling. Interventions that fit these criteria (i.e., they are frequently needed, but the details of the intervention are not very context dependent) are good candidates for being turned into quick recommendations.

Leave initiative to the facilitator. Application log data can provide a rich window into the activities of students, but it cannot fully capture the individual personalities, skills, and finer social dynamics of the workshop environment. Thus, the system should recommend, but not initiate, interventions to students. The system should ensure that system-mediated interventions can be supplemented with the facilitator’s expertise and trained abilities to account for individual students’ personalities, abilities, and other contextual factors.

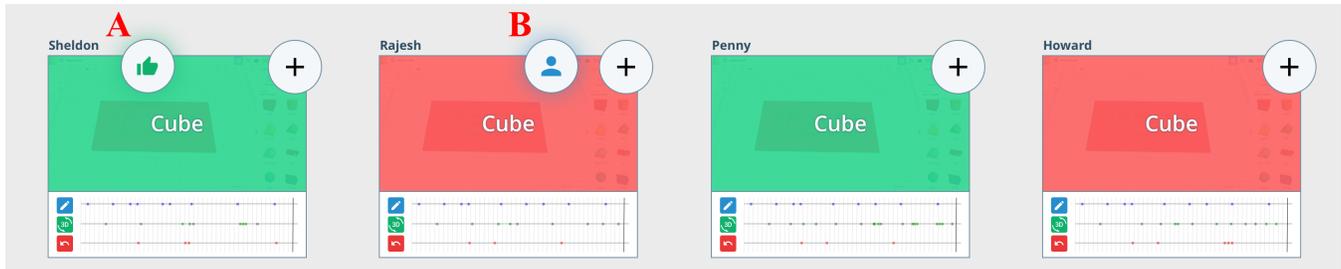


Figure 3. Maestro indicating which students have followed the instructor’s demonstration (in green) and who hasn’t followed yet (in red). (A) A recommended “encouragement” intervention; (B) a recommended “peer help” intervention.

MAESTRO

Based on the general principles described in the last section, we created Maestro, an orchestration system for 3D modeling workshops. Maestro is designed to support workshops on Tinkercad, a popular 3D solid modeling tool often used as a first introduction to 3D design. When describing the individual features of the system, we also include perspectives from two experienced workshop facilitators (one male, one female) that we consulted at the later stages of designing and developing the prototype system.

The primary interface to the system is an interactive dashboard that gives the workshop facilitator an overview of each student in the classroom (Figure 1). Each student is represented by an interactive widget that provides information on that student’s progress (Figure 2). Widget arrangement can be customized by the instructor to, for example, create a representation that mimics the class’s physical seating layout.

Tracking students’ live activity

Maestro enhances the facilitator’s awareness with detailed information on each student’s recent activity. Within each student’s widget, Maestro provides a live view of the student’s application (Figure 2A), as well as a timeline of their recent command history (Figure 2B). Together, the live view and the timeline provide the instructor with the student’s current state and surrounding context. P1 discussed the importance of high-level overviews of current state in managing groups of children, in terms of maintaining “glanceable” awareness of classroom status and being able to quickly note undesirable states:

I like that timeline idea – being able to glance and see if people are on task. [...] Even if I don’t need to act on it immediately at all times. It is useful information as an educator to know what’s going on in a class like that. [...] A lot of it is about managing focus for kids. Trying to keep them on task.—P1

In displaying recent history, we faced the challenge of how much application data to provide, and how to make it easy to interpret by the facilitator. Based on past work on feature-rich software [1], and insights from the workshop facilitators, we settled on the three activity timelines as shown in Figure 2B. The top-most timeline depicts interaction actions (blue), the middle timeline depicts 3D navigation actions (green), and the bottom-most timeline displays undo/erase actions (red). Timelines are updated in real-time, and show the past 2 minutes of activity.

One facilitator was particularly enthusiastic about the ability to see when students were not performing 3D navigation:

Kids are so used to working in two dimensions and on a computer screen. It is still in two dimensions and they are not using the 3-dimensional tools. So, a lot of the times they’ll create something, and when I go to check on it, we’ll find out that nothing is actually attached and in one perspective it looks attached. So, this is a cool tool to me because I can see right away that they’re not using the 3-dimensional tools and as an instructor that would cause me concern. – P2

The instructor can scrub the timeline to replay screenshots of the student’s recent activity, providing further context around events in the timeline. P1 describes how context is necessary for the undo/erase timeline, to disambiguate challenges from creative experimentation:

[A bunch of undo operations] would not always indicate a problem, so I would want to verify [...] I see this all of the time when I’m teaching sculptures. They just undo and undo and undo, and then start doing more work. – P1

Supporting follow-along tutorials

Maestro combines log data from the facilitator’s Tinkercad application, and those used by each student, to identify which students have successfully completed a tutorial step and which have not. When a command is invoked by the instructor (e.g., creating a box), the system monitors for occurrences of that command in students’ logs. If more than 50% of the class performs that command within 60 seconds, Maestro displays ‘follow’ statuses for each student (Figure 3). The ‘follow’ status displays the name of the command, and colors each student’s widget according to whether the student has issued the command (green if they have, red if not). If a student who had not performed the command then performs it, their widget changes from red to green. This gives facilitators a glanceable indication of when it is safe to proceed with the next tutorial step, and can help to identify which students are having trouble following the tutorial instructions.

The workshop facilitators described at-a-glance awareness as being important because of the critical nature of these follow-along exercises, and for assessing the efficacy of their own instruction:

Like a lot of the times the problem is not with the kid, it is with the instructor. The instructor needs to provide the information differently. So, when you notice that you’ve got a fair number of kids that are not getting what you just said, the instructor should maybe be prompted to do something differently. – P2

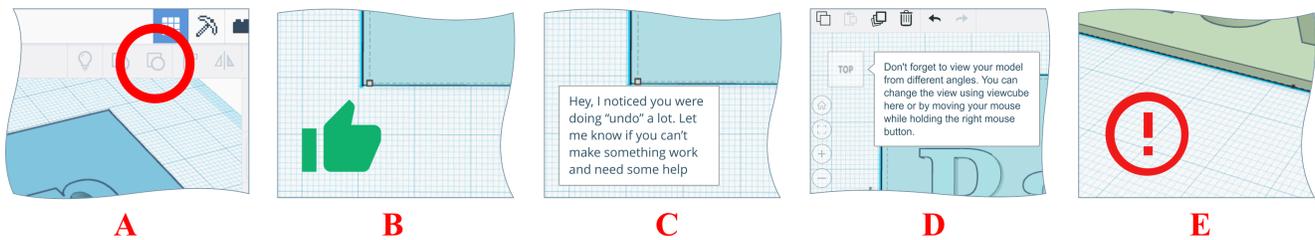


Figure 4. Interventions as they appear in the student’s Tinkercad window. (A) “point to” intervention; (B) “thumbs up” intervention; (C) “text message” intervention; (D) “3D navigation reminder” intervention; (E) “attention” intervention

Detecting common student challenges

Maestro uses heuristics to automatically detect activity patterns that indicate a student is stuck, or is at risk of making a common error, and reports these to the instructor. To help the instructor act upon the detected issues, Maestro also pre-authors and recommends interventions that the instructor can choose to send to the student. The desire for pre-authored interventions, as opposed to a general system for sending any message, was noted in our participatory design process:

Typing is an intensive side task, to stop and type a message [...] There are not a lot of scenarios where you would need to type a custom message. I feel like there are only a few simple messages you would need to send to someone as a reminder or as a further instruction. – P1

Workshop facilitators also spoke to the importance of delivering interventions discretely, to respect the social dynamics present in the classroom:

The benefit is it can act as just a quick efficient reminder without [...] embarrassing a student, or a student feeling called out or unduly criticized. A lot of kids are sensitive. – P2

Maestro’s intervention recommendations appear as large and easy-to-click buttons on the edge of a student’s widget (e.g., Figure 1A,F). In keeping with our design recommendation of leaving the initiative to the expert facilitator, the facilitator can choose to accept the recommended intervention, ignore it, or to otherwise act in response to the recommendation, outside the system. If the facilitator chooses to send an intervention, it appears on the student’s screen as an icon or pop-up, depending on the intervention (Figure 4). Each activity pattern and intervention type is described below.

Encouragement for students following tutorial instructions

Prior work suggests that children in 3D design workshops can benefit from validation or recognition [31]. To support this practice, Maestro recommends an ‘encouragement’ intervention to students who are following the follow-along instruction faster than the classroom average (Figure 3A). This intervention appears as a green ‘thumbs up’ icon in the corner of the student’s Tinkercad window (Figure 4B).

Detecting feature underuse and overuse

Maestro uses real-time log analysis to detect common activity patterns that suggest a student may be struggling, or is working in a way that could lead to later challenges. Specifically, Maestro identifies intervals of time in which students have performed no 3D navigation (Figure 1B). For this pattern, Maestro recommends a ‘3D navigation reminder’

intervention (Figure 1A). If sent by the instructor, this intervention displays a reminder message near the 3D navigation widgets in the student’s Tinkercad interface (Figure 4D).

During the participatory design sessions, the facilitators we interviewed emphasized the importance of keeping these kinds of interventions simple, to avoid distracting the participants from their task:

Most kids that’s all they need – just a reminder. So, this would act as a great little quick reminder. So, I think it would be a positive way to send that reminder, and it’s not shaming, it’s not attacking. – P2

In addition to identifying intervals of time without 3D navigation, Maestro highlights repeated consecutive undo/erase operations on the timeline (Figure 1D), which past work has shown can indicate potential usability problems [1].

Detecting inactivity

Maestro also provides a simple indicator of whether a student has stopped interacting with Tinkercad for an extended period. Lack of activity might indicate that a student became stuck or distracted, and the facilitators we interviewed expressed a strong desire to be aware of such instances:

Just to be able to see at a glance that they are spaced out or staring off into the ceiling kind of thing. Not that you shouldn’t be able to. But anything to make me aware that the student is not [progressing] – P1

In response to this pattern, Maestro marks the student’s widget with a ‘Z-z-z-z’ status (Figure 1E) and recommends an ‘attention’ intervention (Figure 1F). If the instructor sends this intervention, an exclamation point icon appears in the lower left corner of the student’s window (Figure 4E).

Asking peers to help struggling students

To harness children’s social tendencies and willingness to provide peer help [34], Maestro recommends a ‘peer help’ intervention be sent to students who takes more than the classroom-wide average time to follow the instructor’s demonstration (Figure 3B).

If the instructor decides to send the peer-help intervention, Maestro asks the instructor to identify a peer helper, but also assists by recommending the three students who have been most successful in following tutorial instructions so far (Figure 1C). Our current implementation determines success based on the speed with which students have followed the instructor’s follow-along tutorial actions, but more sophisticated approaches could also be explored.

When a student receives a peer-help intervention, a message is displayed in the lower left corner of their Tinkercad window saying that the instructor has asked another student to help them. The student chosen as the helper gets a similar message, asking them to go and help their fellow student.

Manual interventions

Maestro also allows instructors to send interventions to students manually, without waiting for a system recommendation. Currently the instructor can send one of four interventions manually: point-to, thumbs up, text message, and peer help (Figure 2D). The ‘thumbs up’ sends the encouragement intervention described earlier. The point-to and text message interventions provide the instructor with additional ways to send simple messages to students.

Point-to intervention allows the instructor to point to a specific area of a student’s screen. Invoking the point-to intervention brings up an expanded view of the student’s Tinkercad window on the facilitator’s screen. Clicking a point in this window briefly displays a circular indicator on that student’s screen (Figure 4A). This feature was included after input from one of our participatory design sessions:

Being able to highlight portions of the interface [...] so that people can see very clearly on their own screen what I am talking about. That would be really helpful, to get some of the first-time kids to use that stuff, to try it. – P1

Instructors can use this intervention as a simple way to answer common student questions about location of specific interface elements, or to draw attention to specific elements of the 3D model the student is working on.

Text message intervention allows the instructor to send text messages to students, to enable custom interventions beyond those listed above.

Implementation

Maestro works with an instrumented version of Tinkercad that sends activity logs to a server. However, our approach does not rely on deep modifications to the Tinkercad source code, and could easily be extended to other JavaScript-based software by including a simple logging module and adding calls at the point of command invocations. All other features, such as taking screenshots of student’s Tinkercad window, and injecting code into the Tinkercad page to display interventions, are accomplished through a custom-developed browser extension. This separation of concerns makes it easy to extend our approach to other web-based software.

The dashboard interface was implemented as a web application using Angular and RxJS. The server was created using NodeJS, Socket.io, and the Express framework.

WORKSHOP CASE STUDY

We evaluated Maestro in a 50-minute Tinkercad workshop, with an experienced facilitator and 13 participants. The goal of this case study was to get an experienced instructor’s feedback on our approach of data-driven real-time assistance in the context of real workshop, and to gather log data to investigate the effectiveness of our heuristics.

Participants

We recruited 13 volunteers (9 male, 4 female) between 10 and 17 years old using snowball sampling. To run the workshop, we invited one of the workshop facilitators who had provided feedback during the later stages of our system design process. The facilitator had extensive experience running 3D design and printing workshops using Tinkercad. Participants were given a \$25 gift card, and the facilitator was given a \$100 gift card as thanks for her time.

Workshop Setup

Each workshop participant had a computer or a laptop. The facilitator’s Tinkercad screen was mirrored on a large display that was visible to all participants. Maestro was running on a separate pen-based tablet computer, which could be carried around by the facilitator during the workshop.

Procedure

Our study consisted of a 50-minute long Tinkercad workshop, and a post-workshop semi-structured interview with the workshop facilitator. Before the workshop began, we gave the facilitator a brief tutorial on the Maestro system, and walked her through its features. At the start of the workshop, we spent five minutes introducing the participants to the various messages they might receive from the instructor.

Immediately following the workshop, we conducted a semi-structured interview with the facilitator. During the post-study interview, we asked her to give us her impressions of Maestro and its features, after having used the system in the context of a live workshop. We also asked her to compare her experience using Maestro with her prior experience of running similar workshops without such a system.

Participants were also given a short survey about their experience in the workshop, and their perceptions of the utility of the different types of interventions.

Data collection and analysis

The interview was recorded and fully transcribed. Data from the transcript were analyzed by creating affinity diagrams using a bottom-up inductive approach [14]. From these affinity diagrams, we held joint data interpretation sessions among the paper authors to extract common themes. We also collected log data from Maestro and the individual Tinkercad instances, which we analyzed for noteworthy patterns.

RESULTS

Workshop overview and Maestro usage

The facilitator spent the first half of the workshop giving a follow-along tutorial on how to create a personalized keychain in Tinkercad. She started by introducing the tools required for the task, then walked through the steps to create the keychain, pausing after each to allow students time to perform the step. During the follow-along portion of the workshop, Maestro was running on a tablet next to the computer she was using to demonstrate instructions in Tinkercad.

In the second half of the workshop, the instructor asked participants to work on an individual assignment: to use the

tools they had learned to create a cylinder-shaped box with a matching lid. During this phase of the study, the instructor walked around the classroom with the tablet running Maestro in her left arm, so she could check the screen as needed.

Over the course of the workshop, the facilitator sent 40 interventions in total (17 Attention, 10 3D navigation reminders, 8 Point-to, 5 Encouragement). She did not use the peer help intervention during the workshop. In the post-study interview, she mentioned that the peer help intervention would become useful after working with the same group of students over multiple classes.

Command Log Analysis

In this section we examine the log data gathered during the study, to gain insights into the dynamics of in-person group workshops, and how such data can be used to inform data-driven assistance techniques.

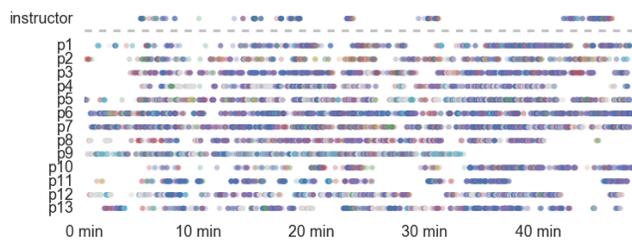


Figure 5. Workshop command log data, showing the instructor’s activity (top) and that of participants (below). Each command is assigned a unique color.

A timeline of all logged command invocations is shown in Figure 5. The facilitator used 20 unique commands, collectively making 317 command invocations. In comparison, the average student invoked 25 unique commands (SD 5), with 1095 invocations (SD 336). The pattern of the facilitator performing fewer commands (and using fewer unique commands) than students is not surprising, given that she had to split her time between presenting a tutorial, checking students’ progress, and assisting individual students. It also suggests that students experimented with functionality not demonstrated by the facilitator.

Follow-along detection

To evaluate our heuristic for detecting instances where students were following the facilitator’s directions, we examined the log data for clusters of students performing a given command at close to the same time. We also applied our existing heuristic to the gathered log data, to see where it succeeded, and where it missed some of these patterns.

The follow-along feature was activated 17 times over the workshop session, across 9 unique commands. Examining these instances in the log data, we generally see distinct clusters of participant command invocations around the instructor’s invocation, suggesting the heuristic was effective in identifying instances of participants following the instructor’s actions. However, we also see evidence of follow-along behavior that was not caught by our heuristic, and a closer examination of some of these clusters suggest some

ways that our heuristics could be improved. Two example clusters are shown in Figure 6.

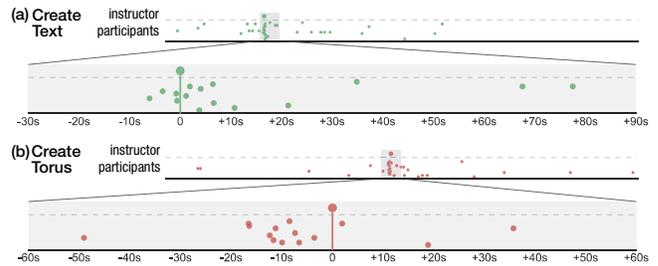


Figure 6. Example clusters of command invocations, for three commands. Vertical lines indicate the instructor’s invocation of the command (where applicable).

For the ‘create text’ command (Figure 6a), our heuristic triggered as expected, but four participants performed the command slightly before the instructor’s invocation. In a more extreme case, for the ‘create torus’ command (Figure 6b), most participants performed the command before the instructor, preventing the heuristic from triggering. Reviewing a video recording of the workshop, we found that the instructor verbally recommended that participants create a torus about 20 seconds before demonstrating how to do so. In addition to the examples shown in Figure 6, for the ‘create cylinder’ tool we observed a cluster of invocations by participants, with the facilitator never using the tool herself. This corresponded with her verbally asking participants to try and “create a cylinder-shaped box”.

Based on these observations, we believe our heuristic could be expanded to better capture follow-along instances. An initial improvement would be to consider the ~30 seconds before the facilitator invokes a command, to catch students who anticipated the step or knew to perform it based on the facilitator’s verbal directions.

3D navigation recommendations

To evaluate the effectiveness of the 3D navigation intervention, we examined participants’ 3D navigation behavior, instances where this intervention was recommended by the system, and instances where the instructor sent this intervention to participants (Figure 7).

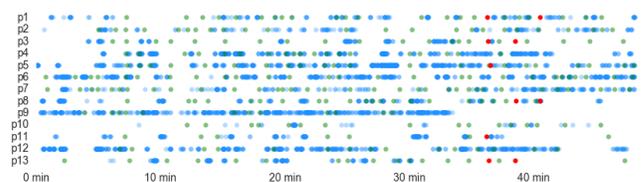


Figure 7. Timeline of participant 3D navigation (blue), recommended 3D navigation interventions (green), and 3D navigation interventions sent by the instructor (red).

All logged instances where this intervention was sent were near the end of the study, while participants were working on an open-ended challenge activity. In terms of prompting the participants to use 3D navigation, the results are promising – in 6 of the 10 instances logged, participants did use the 3D

navigation shortly after receiving the intervention. While this result is encouraging, we note that the instructor sent this intervention only a small number of times in total, and more rigorous evaluation is needed for this feature.

Facilitator impressions

In this section, we discuss the qualitative feedback provided by the workshop facilitator, revisiting our design goals presented at the start of the paper.

Glanceable awareness of student progress

Overall, the facilitator appreciated Maestro's assistance, particularly for the follow-along section of the workshop. She emphasized the utility of the automatic indicators of which students had followed her most recent instruction, and the ability to send students quick affirmations that they were on the right track. We discuss each of these in turn.

The instructor expressed that during follow-along tutorials she usually has very little information on how many students have followed her instructions. She was enthusiastic about the real-time follow-along detection feature, as it allowed her to see at a glance whether participants had followed her directions and were ready to move to the next step:

I really liked the function where I could see if they followed my direction. I thought that was great [...] the screens lighting up green and red [...] It just draws my attention to a really quick glimpse sort of way... who's with me and who is not. [...] I don't have to rely on them to let me know; I have something that shows me that they did not get it. And then I won't speed ahead to the next thing.

The facilitator reported that she traditionally relies on verbally prompting students to tell her whether they have completed the step and are ready to move on. However, she expressed that students often have difficulty assessing their own progress – they may be hesitant to admit that they have run into difficulty, or may be uncertain about whether they followed an instruction correctly:

...there are students who are very quiet and they seem so well-behaved, but they don't express their needs and when you are teaching you have a million things going on in a classroom, it's great that it's something that shows me that there's one student that is struggling...

Building on this, the facilitator expressed appreciation for the “thumbs up” intervention (which was recommended for students who quickly followed her demonstrated actions). She pointed out that students sometimes complete a step, but are unsure about whether they did everything correctly. These students sometimes keep experimenting and “fixing” something that does not need to be fixed. The instructor saw the act of sending a thumbs up as a way to affirm their success and to ask them to wait further instructions:

a lot of times students think they followed your instruction, but they are not sure if they did. Giving them like quick thumbs up "you got it", is good. then they know they could wait for the next instruction if they don't know [if] they got it [right]... I just see it as a potential to say "you are good, wait."

To support this use case, the instructor recommended that the encouragement intervention be recommended for all students who had completed the step, regardless of whether they had done so faster than the class average.

Awareness of student activities in free-form tasks

The facilitator told us that during free-form workshops she sometimes has difficulty helping students because she is unsure of how the student's actions have led to the current challenge they are facing, and that asking students to describe what went wrong is often counterproductive as they cannot always express the source of difficulty. For this reason, she was enthusiastic about the ability to see each student's detailed action history. She told us that she could see herself replaying a student's history to identify the source of the problem, which would allow her to provide more specific help or corrective instruction, in place of spending time trying to diagnose the student's challenge.

This suggests that it might be valuable to provide automated support for diagnosing the source of difficulties. Such an approach could potentially include features for displaying the operation history for particular parts of a document, an approach explored in previous work [18].

Automatic identification of challenges

One of our key motivations for developing Maestro was that existing classroom management tools can be overwhelming, because they simply provide raw screen sharing data to the instructor. The facilitator validated this assumption, and expressed appreciation that Maestro drew her attention to the most crucial issues, such as when a student has not performed 3D navigation for an extended period:

I liked that it showed me that kids weren't using the orientation tool. And sure enough, when I walked over and looked, they had things aligned funny.

Though these features were appreciated, we believe more needs to be done to cut down on the cognitive load of using the system. We observed that the facilitator had difficulty splitting her attention between standard workshop activities and attending to Maestro, something she later confirmed:

[Paying attention to both the classroom and the tool] was really hard. I felt really torn, my inclination is to just pay attention to the children and the classroom, but I also want to use the tool, because I see its value.

While we believe this is an important issue to address moving forward, some of these challenges may lessen with further experience using the system.

Supplementing the instructor's in-person capabilities

In terms of Maestro's ability to supplement the instructor's capabilities, our interview revealed an unexpected use case. The facilitator told us that it is common to have one or more students who struggle much more than other students in a class. In such cases, she felt obligated to stay close to the struggling student, but this could make it difficult to pay attention to the rest of the class:

...there always are students who need more help than others, always! [...] I like to give my attention to the ones who are really really struggling but I don't like abandoning the rest of the students.

Expanding on this comment, the facilitator expressed that Maestro could allow her to send quick interventions to other students while staying close to those struggling the most.

...instead of me being like "just a second" [and coming over to the other student], I could [just] click on her [widget on the dashboard], click right there [where the tool is], and then she sees it. [...] Then I don't have to be running all the way to the place and I can stay with my really high-need kids and do my quick interventions [with the system].

Along similar lines, the facilitator expressed that she would like to be able to zoom-in on a student that she knew was struggling, to monitor them remotely while walking around the classroom. This use case of focusing on a small number of students who need extra attention is an interesting area for future work on data-driven workshop assistance.

Unexpected Use Cases

The facilitator also highlighted some potential benefits of the system that we did not anticipate or design for. Reflecting on her previous experience with running workshops and classes with kids, she expressed that Maestro could help with managing shy students, and with helping her to overcome personal biases while teaching.

Working with shy students

The facilitator told us that it is common for some students to not express when they are struggling until the point where they are too far behind to easily help. She expressed that Maestro's ability to indicate who is following along with her instructions would allow her to detect these students early in a workshop, so she could provide them with support from the start. She expressed that the inactivity indicator could provide a similar benefit in free-form workshops, as students who become stuck often stop using the software entirely:

If I see that there's [...] no action, like nothing is happening, that to me is a good indicator [that they might be stuck] ... I like that it draws my attention to that.

An interesting area for further design is an explicit mechanism for students to ask for help from the instructor, without telegraphing this to their peers (e.g., an 'I'm stuck' button).

Overcoming personal biases

The facilitator appreciated that the system kept track of which students had followed her instructions most closely, and used this data to recommend helpers to send to a struggling student. She told us that she is often too engaged in teaching the workshop itself to keep track of who the strongest students are in an objective sense. She felt that this often lead her to not recognize the abilities of students who may sometimes be disruptive or difficult to work with, but fundamentally have a strong understanding of the material.

She went on to say that helper recommendations based on an objective metrics, such as following the step-by-step tutorial,

would help her recognize strengths of these students, regardless of her personal biases:

it's so important for us [teachers] to be able to see where our students' strengths are and a lot of times it's hard to see it because we have jaded ideas about certain students, if they caused us a lot of grief or whatever. So, how cool it would be if I could see that [student who is often in trouble] is doing exceptionally well and she could be a perfect person to send to [help] another student.

This is an interesting possibility because past research has indicated that strong students sometimes misbehave because they are bored [6]. Harnessing the talent of these students could help create a more interesting and valuable experience for them, while also helping other students in the workshop, and freeing up the facilitator's time and attention as well.

Student Feedback

Questionnaire feedback from the students indicated that they were enthusiastic about the workshop, rating their enjoyment level as 4.2 on a 5-point scale (SD 0.8). Survey responses also suggest that the interventions were understood, and provided a personalized channel for receiving feedback from the facilitator. For example, all participants appreciated receiving the "thumbs up" intervention (M 4.1, SD 0.5), as it gave them assurance that they were on the right track while following the facilitator's instructions:

[It] lets you know you're on the right page or they like what you're doing. (P6)

Attitudes towards the interventions with a more critical component, such as "point to" and "attention", however, ranged (M 3.8, SD 1.0 for "point to" and M 3.4, SD 0.5 for "attention"). Many participants felt that this type of timely, targeted feedback would enhance their workshop experience. Others, however, indicated that badly-timed intervention messages from the instructor might "spoil the fun" for students who want to figure out the problem by themselves, or might "frustrate students with anxiety".

While our focus in this study was on assessing the utility of Maestro for the instructor, a more in-depth investigation of how the interventions impact student experience is a key area for future work.

DISCUSSION AND FUTURE WORK

Overall, our case study validated our design goals for Maestro, and provided encouraging support for the idea of data-driven assistance for group workshops on 3D modeling software. In this section, we discuss opportunities for further developing this approach.

Advanced learner modeling for group workshops

Our system used straightforward heuristics to detect whether a participant has followed a facilitator's instructions, to identify participants who may be experiencing specific learning challenges, and to model the student's ability with the workshop material. This approach was suitable to gain insights into the experience that this type of system can provide to instructors, but more sophisticated techniques could be developed for each of these automated detection capabilities.

As a first step in this direction, it would be valuable to investigate how existing learner modeling approaches (e.g., [4,13]) could be adapted to the real-time workshop setting.

Advances in learner modeling could also help to reduce the cognitive load difficulties that facilitators face when working with multiple students in a workshop. For example, a more sophisticated model of each student's abilities, coupled with more advanced approaches for recognizing challenges, could be used to prioritize interventions – if the learner model indicates that a student is already aware of the importance of varying the 3D view, recommending a 3D navigation recommendation to that student may be low priority, even if they have not moved the camera for an extended period of time.

It would also be interesting to explore approaches that allow a facilitator to develop their own intervention rules. Inspired by work on programming-by-demonstration (e.g., [25]), the system could allow a facilitator to re-play a workshop they had just taught, highlighting instances where they sent interventions in response to a given pattern of operations by students. This could allow the facilitator to “teach” the system additional intervention rules for certain patterns of student behavior. Such replays could also enable the facilitator to reflect on the efficacy of their own instruction, or help them to identify common challenges faced by students when learning specific software features or higher-level tasks.

Finally, while our intention in this work was to investigate a “facilitator in-the-loop” scenario, there are interesting avenues for future work on the role of greater automation in this type of system. For example, with the facilitator's approval, the system could take on more automated control over time (e.g., after learning which intervention recommendations the facilitator tends to accept).

Student interaction in orchestrated workshops

In our prototype system, only the instructor explicitly interacts with the system, with students providing implicit input through their log data. A potential area for future work is to explore how students can explicitly interact with this kind of system (e.g., through functions added to the interface of the application being taught, or an ability to respond to interventions). Given our finding that shy students can be disadvantaged because they do not ask for help, it is worth investigating whether low-cost mechanisms to signal confusion or ask for help could be beneficial. Similar ideas have been explored for lecture settings, using ‘clicker’ devices to create a more active learning environment [28]. The system could also be augmented with tangible devices distributed throughout the classroom, building on past work that has shown the promise of this approach for enabling distributed cognition in learning environments [36].

Generalizability and Scaling

Maestro was designed to address the specific challenges that arise in 3D modeling workshops. Many of its domain-specific features (such as those surrounding underuse of 3D navigation) play a key role in offloading instructor effort,

which our interviews and case study suggest is a key criterion for the success of this type of system. That said, a number of Maestro's features are likely to generalize to other software, including the activity timeline, live-view of students' screens, and detection of undo/erase events.

It would also be valuable to investigate how our approach could be generalized to other class sizes, age groups, and educational contexts. Scaling to larger class sizes is a general challenge for in-person workshops, particularly those with children or novice learners. Typically, as workshops grow, additional in-person support is needed in the form of TAs or additional facilitators. To facilitate these practices, it would be interesting to investigate how our approach could be modified to support simultaneous use by multiple workshop staff.

Generalizability beyond children is also an interesting question. We focused on children as a user group because our formative design process was targeted to this population, but prior work suggests that adults also experience issues learning 3D modeling [21]. We believe many of our system's features, such as its support for follow-along tutorials, would be useful in adult workshops. However, some interventions might need to be re-designed slightly to be acceptable to adult learners, and the social and instructor-student dynamics of workshops with adult learners would need further study.

Finally, we see potential to generalize this approach to remote learning scenarios, to support learners whose local communities do not offer in-person workshops. These scenarios would require developing social sharing features that could remotely support the types of social interactions that make the in-person workshops such a dynamic experience.

Limitations

Future research should expand upon the results of our initial case study. We included 14 participants in our case study (an experienced facilitator and thirteen children), but additional data is needed to validate Maestro's approach across a wider range of workshop styles, facilitator approaches, and classroom dynamics. A multi-session evaluation would also provide deeper insights into the system's awareness and intervention features, once the facilitator had become more familiar with using the system. Finally, the nature of this workshop, which was both introductory and with children who did not know each other, did not allow us to investigate the utility of the peer help functionality in depth.

CONCLUSION

This work has provided initial validation for a data-driven approach to orchestrating workshops on 3D modeling software. Through a prototype system and case study, we have demonstrated the value in techniques that help facilitators to maintain awareness of their class, and assist with identifying and addressing common challenges faced by students. We see the insights from this work as a first step toward a future where workshop facilitators confidently share the effort of teaching with a trusted system, allowing them to focus on creating enjoyable and effective learning experiences for their students.

REFERENCES

1. David Akers, Matthew Simpson, Robin Jeffries, and Terry Winograd. 2009. Undo and Erase Events as Indicators of Usability Problems. *Proceedings of the International Conference on Human Factors in Computing Systems*
2. John R. Anderson, Albert T. Corbett, Kenneth R. Koedinger, and Ray. Pelletier. 1995. Cognitive Tutors: Lessons Learned. *Journal of the Learning Sciences* 4 (2), 167–207
3. Arnold E. Kimberly and Matthew D. Pistilli. 2012. Course Signals at Purdue: Using Learning Analytics to Increase Student Success. *Proceedings of the International Conference on Learning Analytics and Knowledge*, 267-270
4. Ryan Baker, Sujith Gowda, Albert Corbett, and Jaclyn Ocumpaugh. 2012. Towards Automatically Detecting Whether Student Learning is Shallow. *Intelligent Tutoring Systems*, 444-453. Springer Berlin/Heidelberg
5. Ryan S. Baker. 2016. Stupid Tutoring Systems, Intelligent Humans. *International Journal of Artificial Intelligence in Education* 26 (2), 600–614
6. Ryan S. Baker, Sidney K. D’Mello, Mercedes T. Rodrigo, and Arthur C. Graesser. 2010. Better to be Frustrated than Bored: the Incidence and Persistence of Affect During Interactions with Three Different Computer-based Learning Environments. *International Journal of human-computer studies* 68 (4), 223–241
7. Ryan S. Baker and Kalina Yacef. 2009. The State of Educational Data Mining in 2009 : A Review and Future Visions. *Journal of Educational Data Mining* 1 (1), 3–16
8. Je Beck and Jack Mostow. 2008. How Who Should Practice: Using Learning Decomposition to Evaluate the Efficacy of Different Types of Practice for Different Types of Students. *Intelligent Tutoring Systems*, 353-362
9. William Billingsley and Paul Kwan. 2016. Indirect Interaction : A Computing Lecture for Five to Seven Year-Olds. *Proceedings of the Conference Companion Publication on Designing Interactive Systems*, 145–148
10. Zach Broderick, Cristine O’Connor, Courtney Mulcahy, Neil Heffernan, and Cristina Heffernan. 2011. Increasing Parent Engagement in Student Learning Using an Intelligent Tutoring System. *Journal of Interactive Learning Research* 22(4), 523-550
11. Hsiang-Ting Chen, Tovi Grossman, Wei Li-Yi, Ryan M. Schmidt, Björn Hartmann, George Fitzmaurice, and Maneesh Agrawala. 2014. History Assisted View Authoring for 3D Models. *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2027–2036
12. Sharon Lynn Chu, Francis Quek, Sourabh Bhangaonkar, Amy Boettcher Ging, and Kumar Sridharamurthy. 2015. Making the Maker: A Means-to-an-Ends Approach to Nurturing the Maker Mindset in Elementary-Aged Children. *International Journal of Child-Computer Interaction* 5, 11–19
13. Cristina Conati, Abigail Gertner, and Kurt Vanlehn. 2002. Using Bayesian Networks to Manage Uncertainty in Student Modeling. *User Modelling and User-Adapted Interaction* 12(4), 371–417
14. Juliet Corbin and Strauss Anselm. 2008. Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. SAGE Publications Inc.
15. Daniel Fitton, Janet C. Read, and John Dempsey. 2015. Exploring Children’s Designs for Maker Technologies. *Proceedings of the International Conference on Interaction Design and Children*, 379–382
16. Adam Fourney and Ben Lafreniere. 2014. InterTwine: Creating Interapplication Information Scent to Support Coordinated Use of Software. *Proceedings of the ACM Symposium on User Interface Software and Technology*, 429–438
17. Leonel Morales Díaz, Laura S Gaytán-lugo, and Lissette Fleck. 2015. Profiling styles of use in Alice: Identifying patterns of use by observing participants in workshops with Alice. *Blocks and Beyond Workshop*, 2015 IEEE, 19–24
18. Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2010. Chronicle: Capture, Exploration, and Playback of Document Workflow Histories. *Proceedings of annual ACM Symposium on User Interface Software and Technology*, 143–152
19. Julio Guerra, Roya Hosseini, Sibel Somyurek, and Peter Brusilovsky. 2016. An Intelligent Interface for Learning Content: Combining an Open Learner Model and Social Comparison to Support Self-Regulated Learning and Engagement. *Proceedings of the International Conference on Intelligent User Interfaces*, 152–163
20. Philip J Guo. 2015. Codeopticon: Real-Time, One-To-Many Human Tutoring for Computer Programming. *Proceedings of the ACM Symposium on User Interface Software & Technology*, 599–608
21. Nathaniel Hudson, Celena Alcock, and Parmit K Chilana. 2016. Understanding Newcomers to 3D Printing: Motivations, Workflows, and Barriers of Casual Makers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*: 384–396
22. Nathaniel Hudson, Benjamin Lafreniere, Parmit K. Chilana, and Tovi Grossman. 2018. Investigating how online help and learning resources support children’s use of 3D design software. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 10

23. Judy Kay, 1997. Learner Know Thyself: Student Models to Give Learner Control and Responsibility. *International Conference on Computers in Education*, 17–24
24. Ben Lafreniere, Tovi Grossman, and George Fitzmaurice. 2013. Community Enhanced Tutorials: Improving Tutorials with Multiple Demonstrations. *Proceedings of International Conference on Human Factors in Computing Systems*, 1779–1788
25. Tessa A. Lau and Daniel S. Weld. 1998. Programming by Demonstration: An Inductive Learning Formulation. *Proceedings of the International Conference on Intelligent User Interfaces*, 145–152
26. Wei Li, Tovi Grossman, and George Fitzmaurice. 2014. CADament: a Gamified Multiplayer Software Tutorial System. *Proceedings of the International Conference on Human Factors in Computing Systems*, 3369–3378
27. Chee Kit Looi, Wenli Chen, and Foo Keong Ng. 2010. Collaborative Activities Enabled by GroupScribbles: An Exploratory Study of Learning Effectiveness. *Computers and Education* 54(1), 14–26
28. Zeny F Mateo. 2010. Creating Active Learning in a Large Introductory Statistics Class Using Clicker Technology. Data and context in statistics education: Towards an evidence-based society. *Proceedings of the International Conference on Teaching Statistics*
29. V. R. Raghuvver, B. K. Tripathy, Taranveer Singh, and Saarthak Khanna.. 2014. Reinforcement learning approach towards effective content recommendation in MOOC environments. *International IEEE Conference on MOOC, Innovation and Technology in Education*, 285–289
30. Susan Bull and Judy Kay. 2010. Open learner models. *Advances in intelligent tutoring systems*, 301-322
31. Thomas H Park, Rachel M Magee, Susan Wiedenbeck, and Andrea Forte. 2013. Children as Webmakers : Designing a Web Editor for Beginners. *Interaction Design and Children*, 419–422
32. Suporn Pongnumkul, Mira Dontcheva, Wilmot Li, Jue Wang, Lubomir Bourdev, Shai Avidan, and Michael F. Cohen. 2011. Pause-and-Play: Automatically Linking Screencast Video Tutorials with Applications. *Proceedings of the ACM Symposium on User Interface Software and Technology*, 135–144
33. Gabriela T. Richard and Yasmin B. Kafai. 2015. Making physical and digital games with e-textiles. *Proceedings of the International Conference on Interaction Design and Children*, 399–402
34. Oliver Stickel, Dominik Hornung, Konstantin Aal, Markus Rohde, and Volker Wulf. 2015. 3D Printing with marginalized children—an exploration in a Palestinian refugee camp. *Proceedings of the European Conference on Computer Supported Cooperative Work*, 19-23
35. Keith J. Topping. 1996. The Effectiveness of Peer Tutoring in Further and Higher Education: A Typology and Review of the Literature. *Higher Education* 32(3), 321–345
36. David Verweij, Saskia Bakker, and Berry Eggen. 2017. FireFlies2: Interactive tangible pixels to enable distributed cognition in classroom technologies. In *Proceedings of the ACM Conference on Interactive Surfaces and Spaces*, 260–269
37. Juan Diego Zapata-Rivera and Irvin R. Katz. 2014. Keeping Your Audience in Mind: Applying Audience Analysis to the Design of Interactive Score Reports. *Assessment in Education: Principles, Policy & Practice* 21(4), 442–463