

Exploring Hypotheses about Media Computation

Mark Guzdial
School of Interactive Computing
Georgia Institute of Technology
85 5th Street NW
Atlanta, Georgia USA 30332-0760
guzdial@cc.gatech.edu

ABSTRACT

Research in computing education has been criticized as “Marco Polo,” e.g., the researchers tried something and reported what happened. Our developing field needs more hypothesis-driven and theory-driven research. We will get there by making clear our goals and hypotheses, testing those goals and hypotheses explicitly, and critically reconsidering our results. My colleagues and I designed and evaluated a media-centric introductory computing approach (“Media Computation”) over the last ten years. We started from a “Marco Polo” style and an explicit set of hypotheses. We have worked to test those hypotheses and to understand the outcomes. Our iterative effort has led us to explore deeper theory around motivation and learning. This paper serves as an example of a ten year research program that resulted in more hypotheses, a more elaborated theory, and a better understanding of the potential impacts of a computer science curriculum change.

Categories and Subject Descriptors

K.3.2 [Computers and Education]: Computer and Information Science Education — computer science education

General Terms

Education

Keywords

Assessment, curriculum, education research, motivation, retention, broadening participation, women, under-represented minorities

1. SETTING HYPOTHESES

Valentine critiqued computer science education research papers in 2004 [45] as having a “Marco Polo” style: “I went there and I saw this.” He pointed out that such papers play an important role in communicating experience to fellow

educators, but did not have the same value as experimental papers, which conducted some theory-driven analysis of a given treatment. In his meta-analysis, he characterized the majority of papers published in the SIGCSE conference were “Marco Polo” papers. I suggest in this paper that “Marco Polo” is a natural condition at the beginning of a research project. Explicit statement of hypotheses, testing against those hypotheses, and iterating to refine and integrate other results and theories creates a progression to producing theory.

Since 2002, my students and I have been designing and evaluating an approach to teaching introductory computation with a media-focused context, named *Media Computation* (or *MediaComp*). MediaComp introduces computing through exploration of data abstraction related to digital media. Students manipulate pixels to create Photoshop-like image effects, samples to splice or reverse sounds, text to compose or search HTML pages, and frames in a video. At Georgia Tech, MediaComp is taught using Python, but at other institutions (such as UCSD [37]) they teach similar things in Java. Our first paper which proposed the approach was *even less* of a research contribution than a “Marco Polo” paper, since we merely considered the possibility of a media-centric approach [25]. The first two papers describing the initial implementation were clearly “Marco Polo” style papers [26, 17] describing what we were trying and some of the initial responses from the students.

In addition to the reports of “we went there and saw this,” those two published papers (the ITICSE 2003 paper [17] and our first design paper [24]) and an internal design document for the course [16]¹ together lay out a series of design goals. The evaluation efforts that followed were explicitly testing those design goals as hypotheses, where the hypotheses were that the implementation of the course succeeded at the design goals. These hypotheses have been tested and explored by my students and me over the last ten years. In this paper, I also draw on three studies of the use of MediaComp at other institutions, to attempt to generalize the results across institutions.

This paper is a retrospective consideration of the exploration of the MediaComp hypotheses, to describe the growth of the effort from “Marco Polo” to theory-driven and theory-creation. The central hypotheses are:

- **The Plagiarism Hypothesis.** Georgia Tech began requiring a course in computer science of all students in 1999, but a crisis involving plagiarism led us to develop

¹Still accessible at <http://coweb.cc.gatech.edu/mediaComp-plan/uploads/1/working-document-v3.html>

MediaComp and two other contextualized courses in 2003 [19]. One course was taken by Engineering students and used MATLAB for all programming. The MediaComp course is taken only by students from the Colleges of Architecture, Management, and Liberal Arts. We hoped that the media focus would be more engaging and relevant for these students, and would reduce the incentive to cheat. An explicit design goal for MediaComp was that “*Academic misconduct cases [will be] brought against less than 5% of the students*” [16].

- **The Retention Hypothesis.** When all students at Georgia Tech took the same introductory programming course, students in programs from our Colleges of Architecture, Management, and Liberal Arts failed or withdrew from the course at a rate of over 50% (i.e., fewer than 50% completed the course with a passing grade) each semester [27][19]. An explicit design goal for the course was for “A drop-failure rate of less than 15%” [16]. To broaden the hypothesis beyond our school, let’s phrase this hypothesis *that a course using Media Computation will have a higher retention rate than a traditional course.*
- **The Gender Hypothesis.** We designed the course drawing on the the recommendations on how to improve computing curricula to be more engaging and successful for female students [31]. An explicit design goal for the course was that “*The enrollment for the course will be at least 30% female, and the drop-failure rate will be no different for females than males.*”
- **The Learning Hypothesis.** We laid out a series of learning goals in our designs for MediaComp [16, 17], but only implicitly suggested that learning would be unchanged between the traditional course that all students had been taking and the new MediaComp course. We said that we would have “a goal of directly measuring learning in CS1315 (the MediaComp course at Georgia Tech) in comparison with other CS1 courses.” Thus, there was an implicit design goal *that students would learn as much in the MediaComp course as in other introductory computing courses offered at Georgia Tech.*
- **The More-Computing Hypothesis.** We never explicitly set a goal that students who took MediaComp would be more likely to major or minor in computer science. In part, that was a political decision. We wanted the support of our colleagues in other Colleges who were helping us create the course, and we did not want them to see us as “poaching” their students. However, we did set a goal (in [24]) that “The course should have impact beyond the single term. If the course doesn’t influence how non-major students think about computing, and they will only take a single computing course (probably), then we will have lost our opportunity to influence these students.” The College of Computing did create a minor in Computer Science in 2004, and we mentioned in that same paper our hope that MediaComp students could pursue the minor and “that students interested in computing can go into more depth without leaving their own majors” [24]. We defined a potential second course in our design document [16] that would enable students to go

on to take more computer science courses. Thus, there was an implicit design goal *that MediaComp students would take more computer science courses.*

Most research papers present results from a single study. Many (but not all) of the MediaComp hypotheses have been explored in one or more studies. In this paper, I aim to tell a broader story about the exploration of a *set* of hypotheses in a research *program*. Across multiple studies, were these hypotheses supported or refuted? What hypotheses did we not explore? How did the hypotheses that we explored change over time?

Our point of comparison at Georgia Tech for the “traditional” class is the single course that everyone at Georgia Tech took from 1999–2003, before we created three separate courses [19]. At other schools, the “traditional” course is the previous incarnation of the course that MediaComp replaced. Each “traditional” course is different at different schools, but there are some generally accepted descriptions of an introductory “CS1” course [2], and there are general expected characteristics about success and failure in these courses [4].

The studies I cite here from outside of Georgia Tech adopted MediaComp with similar goals (e.g., to improve retention rates). Courses that use MediaComp curricula (Python [18], Java [21], and Alice+Java [5]) are not identical. Teachers have different goals for adopting MediaComp. In a study of faculty in professional development, we found that most teachers adopted MediaComp because of the teacher’s personal enthusiasm about the content, and not to improve retention rates or student motivation [33]. Different goals would lead to different implementations. I aim to reduce variance in course implementations by focusing just on those papers with a similar goal, to improve retention.

2. THE PLAGIARISM HYPOTHESIS

The plagiarism hypothesis is unsupported and the least explored of the MediaComp hypotheses. The plagiarism hypothesis has never been explicitly tested at Georgia Tech. In the first semesters of the MediaComp introductory course, the question was not even considered. Students produced programs whose creative output could be easily directly compared with other students’ products. Plagiarism within the course would be easily identified. The course didn’t exist elsewhere, so it would have been difficult for potential plagiarists to find programs to copy.

But over successive semesters, we started to detect cheating. A student would turn in an assignment whose output was identical to one by a student from a previous term or year. We began changing the assignments so that they would be different than past semesters, but that just started the arms race. We have since found MediaComp assignments on programmers-for-hire websites.

We do not have any direct measure of the number of plagiarism (academic misconduct) cases, or of the number of students found guilty. We have asked instructors over the last six years about plagiarism in the MediaComp course. They say that their sense is that they file about as many academic misconduct claims in the MediaComp course as in other introductory courses.

We had hoped that increased engagement and motivation would reduce the incentive to cheat. We have not attempted to test this hypothesis, but the anecdotal evidence suggests

Table 1: Retention data for Georgia Tech’s MediaComp course from Fall 2006–Fall 2012

	Female	Male	Total
Passing grades	2102	1659	3761
Failing grades	208	235	443
Withdraw	30	46	76
DFW Total	238	281	519
% DFW-total	5.5%	6.5%	12%
% DFW-set	10.1%	14.5%	12%

that we were wrong. Whatever incites plagiarism, MediaComp does not seem to impact plagiarism.

3. THE RETENTION HYPOTHESIS

The retention hypothesis is the most explored and best supported MediaComp hypothesis. We originally hypothesized that we would have a failure² rate less than 15%. Our results in the pilot offering of the course showed that less than 12% of students withdrew from the course or had a failing grade (Unpublished internal report[23]³). In fact, the retention rate rose compared to our traditional course in both our MediaComp course and in our Engineering-oriented MATLAB course that started the same semester[14]. The average failure rate for students from the Colleges of Architecture, Management, and Liberal Arts had been over 50% in the two years before MediaComp started, and less than 15% in the following two years [27].

A review of the last six year’s offerings of the Georgia Tech MediaComp course (4,292 total students) show the failure rate continues to be below 15% (Table 1). We use 4,280 students in our calculations because students took an incomplete and finished the course the following semester, and are not considered to be passed or failed. We compute % DFW-total in terms of DFW for men, women, and total over 4280, then %-DFW in terms of DFW over the number within that set. The course at Georgia Tech was majority female because the majors who were required to take it (from the Colleges of Liberal Arts, Management, and Architecture) are more female (than the other colleges at Georgia Tech – Engineering, Computing, and Sciences).

In published accounts of other schools’ MediaComp courses, the retention hypothesis has been well-supported. We studied the first adoption of MediaComp at Gainesville State College, and found an increase in retention over the traditional course [43]. The University of Illinois-Chicago adopted MediaComp for their “CS 0.5” course, and they found a significant improvement in retention [38]. The University of California at San Diego adopted MediaComp in 2008 and document an improvement in retention [37]. Five years later, Porter & Simon presented a longer term analysis on how their adoption of peer instruction, pair programming, and MediaComp in 2008 led to greater retention with the computer science major measured into the second year of undergraduate [34].

²Students who withdraw from the course or earn a failing grade are considered to have ‘failed’ (DFW) in this analysis. The students who earn a passing grade (A, B, or C) are said to have ‘succeeded’ and were ‘retained.’

³<http://coweb.cc.gatech.edu/mediaComp-plan/uploads/45/FINAL-report-pilot-offering.1.pdf>

3.1 Explaining the retention effect

Why did retention improve at multiple schools over several years? There may be multiple causes, and they may be different at different schools. We can imagine differences in grading practices between different MediaComp courses which might explain retention differences, for example. Given that the result was similar across different MediaComp instantiations, and that our goal is to build a theoretical explanation, we will focus on the common curriculum between the classes. How might the MediaComp curriculum have influenced retention?

In our first year of implementing MediaComp, we conducted several interview studies to understand how students saw the course. Students told us that the course was “motivating” and “fun.” Why? Several explanatory themes arose from those studies:

- Students told us that they appreciated that the course was “tailored” to their major, in both the MediaComp and Engineering courses [14].
- Students told us that they found the course to be a welcome opportunity to be creative [24][35].
- The theme of relevance appeared prominently in our findings. Female students told us that they understood what they would do with the content of the MediaComp, but those in the traditional course said that they did not know what they would do with what they were learning [35]. In a follow-on study a year after we started the MediaComp course at Georgia Tech, we found that 19% of the respondents to a survey had actually done some programming outside any course context [24].

Here is a quote that captures the relevance point well [24], where a student describes coding beyond the requirements of the course:

I just wish I had more time to play around with that and make neat effects. But JES (the Python IDE for the MediaComp course) will be on my computer forever, so... that’s the nice thing about this class is that you could go as deep into the homework as you wanted. So, I’d turn it in and then me and my roommate would do more after to see what we could do with it.

We found student reaction to the course surprising when viewed from a perspective of “thick authenticity” [36]. Students found MediaComp relevant and something that they could use for later programming. Yet the course is quite *inauthentic* [27]. Real programmers who manipulate media do not write code like what students do in MediaComp. MediaComp is far too slow to be usable in practical settings. However, students clearly saw that they could do useful tasks with code in MediaComp [24], even if those tasks would not be performed *in that way* and *with those tools* by professionals.

We made a theoretical choice at this point in our exploration of MediaComp, in part driven by the finding that the retention effect occurred in *both* the MediaComp course and the Engineering-focused course. The Engineering course did not make an effort to provide a creative outlet. We decided to focus on the *sense of relevance*. Creativity may be an important part of why MediaComp is motivating to students.

The sense of “tailoring” may also be playing an important role. Given that we had two examples of courses that were having a retention effect, we decided to focus on the common themes between the two courses.

So what led to the sense of relevance? Our hypothesis was that it wasn’t the media, but the use of a consistent application *context*. When we look at the issues that students identified in response to “What do you like the most in this class?” in both the MediaComp and Engineering courses [14], they were issues of application contexts. The MediaComp students liked manipulation of media and programming. The Engineering students liked graphing and data manipulation. (The most common response to that question in the traditional course was “Nothing.”)

Our hypothesis was that other application contexts would also be effective to inculcate a sense of relevance. We explored the use of context to support motivation through a sense of relevance in other courses at Georgia Tech.

- Georgia Tech taught a new kind of computer architecture course where students programmed a Nintendo Gameboy. In a study comparing our traditional computer architecture course with the Gameboy-contextualized course, we found greater motivation among students in the Gameboy course without a significant difference in learning [42].
- We also taught a version of introductory computing using robots. Students learned to program by controlling a small robot using Python [3]. Our evaluation of this effort suggested that robotics was *not* significantly more motivating or having a greater impact on retention than our traditional course [40]. One explanation for this finding is that students did see the *relevance* of the context of robotics, but it did not *motivate* them. Robotics was too hard and confusing to students. Another study of a similar curriculum and robotics platform also found no impact on student motivation, for similar issues around the instability of the robotics platform [32].

Georgia Tech decided that contextualization was a successful strategy for undergraduate learning. Our entire undergraduate curriculum was reorganized around the contextualization. Our “*Threads*” [15] structured courses around application areas.

Does contextualization really help? It does not *always* lead to higher retention or improved motivation[20]. We certainly have evidence of context leading to positive motivation impacts (Engineering, MediaComp, and Gameboy), and evidence where there is no measurable impact on motivation (Robotics). There is strong evidence that context *can* lead to a sense of relevance, but that sense of relevance does not always lead to motivation to succeed in the course. What are the characteristics of a context that leads to relevance and motivation? It may depend on a student’s background, or in the details of how the context is expressed to the student. Clearly, it *can* work, but we do not know how to *make* it work *consistently*.

Figure 1 summarizes our theory of how MediaComp influences retention. Arrows indicate hypothesized directions of influence. We have evidence that MediaComp has an effect of raising retention. We have evidence that the contextualization aspect of MediaComp influences female students

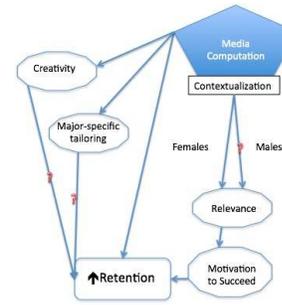


Figure 1: Hypothesized impacts of Media Computation on Retention.

perception of relevance, which motivates success, and leads to greater retention. Question marks are placed on hypotheses for which we have no evidence. We have evidence that MediaComp leads to perceptions of “creativity” and “tailoring,” but we have not explored what the impact of those on other factors such as motivation to succeed and retention. We also know little about the influence of MediaComp on *male* student’s perceptions of relevance since our early interview work was only with women.

3.2 Issues of Relevance beyond Context

Exploring the retention hypothesis led my PhD students and I to explore other related issues. In particular, we were struck by the MediaComp students’ sense of relevance *despite* the inauthenticity of what we were teaching [27]. What would it look like to build a learning opportunity that was truly authentic and relevant? How do students judge relevance?

Brian Dorn explored in his thesis work how to teach computing that is relevant to a subsection of the MediaComp audience at Georgia Tech in an explicitly authentic manner [7, 10, 8, 9]. Brian studied how graphic designers are teaching themselves programming [8, 9]. Few of the graphic designers that Brian studied had computer science or even much mathematics background. Brian identified computer science topics and practices that the graphic designers definitely needed [10], i.e., he identified authentically useful computer science. However, they *explicitly* rejected computer science courses and books. Brian was challenged to find a way to provide learning opportunities that the designers found relevant. Brian supported the graphic designers in learning computer science by providing a case library of useful programming segments which he annotated with computing concepts [11]. When asked to use the case library or an equivalent library of programming segments without annotations, Brian found that the participants using the case-based materials demonstrated growth in their conceptual understanding of CS topics, the library users did not learn about CS, and he found no difference in success in complete programming tasks between the two groups [6]. In short, he could provide authentic and relevant learning opportunities, but outside formal structures like textbooks and classes.

Mike Hewner in his dissertation asked how computer science students make education decisions, with special consideration about the influence of their possibly inaccurate conceptions of the field of computer science [29]. He conducted a Grounded Theory analysis with over two dozen interviews with students from three different institutions. What Mike

found was that students actually have little real understanding of what is authentic practice. Few of the students he interviewed had a clear idea of what kind of job they were preparing themselves for. Mike studied students who were majoring in computer science. How much less might non-CS majors (like those in Architecture, Management, or Liberal Arts) know about what is authentic computing practice even within their own discipline? So students found MediaComp, even though it was inauthentic, because (we theorize) students don't know what *is* authentic practice. In papers on how MediaComp was designed, my students and I suggest that we can *design* a course to create a sense of relevance [27, 24], but we only have the one instance of MediaComp to offer as evidence.

4. THE GENDER HYPOTHESIS

From the beginning of MediaComp at Georgia Tech, we found support for the gender hypothesis. The “at least 30% female” part of the hypothesis was easy. Given the gender distribution in the majors who were required to take MediaComp at Georgia Tech, the course was almost always more than 40% female. What's striking is that the success rate (percentage of originally-enrolled students who complete the course with a passing grade) between the female and male students was very similar [23]. In fact, women often succeeded at a higher rate than men, as can be seen in Table 1 summarizing the last six year's worth of data. Other schools found roughly similar success rates between genders as well [38, 34].

We did two interview studies to explore why women found the class more motivating than the traditional course. Female students in our first study told us that they liked the content more in the MediaComp class [13]. In a larger study in which we interviewed women in both our MediaComp and our traditional course, students emphasized the sense of relevance. The women we interviewed valued that the content was useful [35].

We explicitly designed MediaComp based on recommendations from studies about women's experience in computing courses. We emphasized applicability and usefulness [31], and created opportunities for creative expression [1]. Given the underrepresentation of women in computing, this is a useful result. However, what we don't know is how men responded to the same design features.

The research support for the gender hypothesis lead was a key support for the proposal to establish the NSF-funded alliance to broaden participation in computing, “Georgia Computes!” The strategy for “Georgia Computes!” was to use motivating contexts (like media, story-telling, and robotics) to engage a broader audience of students. “Georgia Computes!” gave us an opportunity to explore a variety of related questions, like how encouragement has a greater positive impact on women and under-represented minority students than on majority males [22].

While the gender hypothesis is supported, and we have explored why women succeeded, we have not thoroughly explored the hypothesis in that we do not know how MediaComp impacts men. We have not explored the qualitative difference in experience between the male and female students, nor how men experience the class. Given that the success/failure rates are about the same, it may be that both genders are experiencing the courses similarly. It might be true that men are actually experiencing the course more *neg-*

atively than the women, and that the men would be more successful in a more traditional course.

The obvious question beyond gender is about the role of MediaComp in supporting under-represented minority students. Georgia Tech has few under-represented minorities. The MediaComp course at University of Illinois-Chicago did have more minorities and did have a significant retention result [38], but we do not have data disaggregated by racial/ethnic groups. We do not have qualitative evidence about how minority students experience the course.

5. THE LEARNING HYPOTHESIS

We had difficulties measuring what was being learned in the three courses from our original pilot [23]. We did put similar problems on the exams of all three courses, but found that the differences in language and sequence made the results incomparable. For example, the Engineering students using MATLAB were very good in reasoning about arrays, but actually were not very good at `for` loops. The use of `for` loops is rare in MATLAB. All three groups of students did badly at the Rainfall Problem, as might have been expected [39].

Allison Elliott Tew conducted a study measuring learning between the three introductory courses at Georgia Tech, presented at the first ICER conference [44]. She developed a test that measured learning outcomes (a) that were common across our three introductory courses and (b) learning outcomes from the second course in our introductory sequence. She gave it to students at the beginning of the second course, and at the end of the second course. At the beginning test, she could clearly distinguish the students between the three different courses. But on the test at the end of the second course, she found no differences between the students. Do any learning differences matter in the first course, if they disappear by the end of the second course?

However, in successive trials, Allison was not able to replicate the result. She started to suspect that the problem was in her instrument. If students were not always reading the problems in exactly the same ways, or if some problems were harder than others (but weighted the same), the scores on the test would be unreliable. She decided that she had to develop a measure of computer science knowledge that she could trust.

Allison's dissertation work led to the creation of the *FCS1*, the test of Fundamental CS1 knowledge [41]. The *FCS1* is the first validated test of introductory computing knowledge that correlates with performance in Java, MATLAB, and Python. With *FCS1*, we could finally compare performance between the three CS1's. She found that the MediaComp students did less well than the students in the other courses.

Finally, we had our answer, but it was not a convincing answer. The MediaComp students did less well, but these were also the students with the least prior computing background [23]. We were comparing students who had chosen majors in Liberal Arts, Architecture, and Management to students who had chosen majors in Engineering, Science, and computing. We could not expect any curriculum to make up for differences in interest and background.

One reason for believing that MediaComp may be impacting learning is because it seems to have an effect on time-on-task. The quote appearing earlier had a student talking about “doing more after” her program. Lana Yarosh found that students in the MediaComp CS2 course did more on

their programs “just because they were fun” [47]. We know that MediaComp students find the curriculum to be creative [35], and Tammy VanDeGrift found that students who find programming creative tend to explore more options with their programs [46].

The learning hypothesis may be poorly defined. We should not expect similar knowledge outcomes between the different introductory courses. Rather, we should have investigated *learning gains* [28]. We could measure *difference* between the students’ knowledge at the start of the course and at the end. An effective learning opportunity supports an increase in knowledge from start to end. A better statement of the learning hypothesis in terms of learning gains is *That MediaComp leads to a greater or equivalent learning gain in comparison with the traditional computing course.*

6. THE MORE-COMPUTING HYPOTHESIS

I believed that MediaComp would lead to more students from Liberal Arts, Architecture, and Management taking computer science classes. Even in the 2002 working document [16], I proposed the creation of a second MediaComp course that would lead students to succeed in the traditional computer science sequence. I created that second course which focused on basic data structures and their role in implementing more advanced digital media products, from linked lists for describing music compositions, to scene graphs for producing animations. Lana Yarosh conducted an evaluation of the MediaComp data structures course [47]. She found that 70% of the class agreed or strongly agreed that working with media made the class interesting, and 67% of the students in the second class agreed or strongly agreed that they were really excited by at least one class project.

At Georgia Tech, the MediaComp introductory course typically has about 300 students each semester, with a total of 700–800 a year. Less than 5% of the MediaComp students ever signed up for the second course. Less than 10% of those students ever pursued a computing minor or a change of major. Even today, ten years later, there are fewer than 250 students total who are pursuing a computer science minor at Georgia Tech. In contrast, there are 1,027 students in the computer science major at Georgia Tech today.

Why didn’t we get more students to pursue computing? Mike Hewner conducted an autobiography study in 2007 [30], four years after we started MediaComp and our Engineering classes. He wanted to understand the impact of a tailored introductory course through the rest of the students’ undergraduate career. He found that the introductory course had little impact in terms of change in attitudes towards computing. If a student was pursuing a major (say, in computing or engineering) in which she expected to use computing, then the introductory course gave useful preparation for that major, and the student enjoyed the course. If a student was pursuing a major (say, in liberal arts or architecture) in which she expected to use *little* computing, then the introductory course was an annoying bump in the road, and it was followed by little computing over the rest of the degree. Thus, the decision about major (which is made on the application at Georgia Tech) is a much greater influence on students’ likelihood to pursue more computing than the introductory course.

In understanding the more-computing hypothesis, my students and I drew on the work of Jacqueline Eccles [12] who studied educational decision-making. She found that a stu-

dent’s choice to pursue a particular major was driven by issues of cultural milieu (e.g., gender stereotypes), the student’s perception of gender roles and activity stereotypes, the student’s goals, and the student’s expectations of success. The student’s choice is also driven by *subjective task value*, e.g., the utility of the choice and enjoyment value. Our curriculum can influence the subjective task value, by making the content more useful and more fun, but the curriculum is not useful for changing the other factors.

7. CONCLUSION: FROM POLO TO THEORY

My work with MediaComp started with “Marco Polo” stories, describing the experience with the then-novel curriculum. We started MediaComp with a set of explicit design goals and hypotheses. Over the following years, we tested these hypotheses, and developed new theories about what was happening and why. In a ten year journey, we moved from “Marco Polo” to theory. We have findings about our original hypotheses, new hypotheses that help us to interpret those findings, and even newer findings about the new hypotheses. We made the transition from “Marco Polo” to generating theory because we iterated on hypotheses, testing those hypotheses, and reflecting on our understanding of the results to generate new hypotheses.

To summarize, we have a set of theoretical positions that we can support and some we can reject. We have good support for the retention hypothesis and for the gender hypothesis. Our results lead us to reject the learning hypothesis and the more-computing hypothesis. We have not tested plagiarism hypothesis, but our personal communications with the teachers suggests rejection.

After ten years, we have more unexplored hypotheses and questions than we started with. These emergent hypotheses and questions drive exploration in pursuit of richer and more detailed theoretical models.

- While the evidence in support of the retention hypothesis is good, I don’t have a convincing explanation for the retention effect. My current hypothesis is that the context leads to a sense of relevance which leads to motivation to succeed in the course. But not all contexts are equally successful. What are the characteristics of a context and the students’ relationship with that context that lead to successful learning? My students and I did not explore all the possible explanations for the retention effect, like the creativity of MediaComp assignments or the value of students having a sense that a class was “tailored” just for them.
- We have suggested that we can *design* for relevance (and thus, retention). An interesting experiment would be to use our process to create a similarly contextualized curriculum, then test if it reliably resulted in higher retention.
- The gender hypothesis is supported. Our work on gender in Media Computation has only explored the issues of women. I do not have evidence about the difference (if any) in how male and female students respond to Media Computation.
- Our continued exploration of the retention and gender hypotheses have raised a set of new issues. Adult students, students from different careers (beyond graphic

designers), and different under-represented minority groups are all potential students who might experience MediaComp or other contextualized courses differently than the students we have studied up until now. How do we engage others?

- The learning hypothesis was not supported, but I have a revised learning hypothesis based on learning gains.

Our experience with MediaComp over the last ten years gives us a better idea of what a curriculum can influence and what it is unlikely to influence. Curriculum can clearly influence motivation to continue in a course and even in a degree program. A curriculum is *unlikely* to have an influence on learning greater than effect of the student's background, choices, and interests. Not everything we might want to change can be influenced through our design of curriculum.

8. ACKNOWLEDGMENTS

My thanks to my collaborators Brian Dorn, Barbara Ericson, Rachel Fithian, Andrea Forte, Mike Hewner, Tom McKlin, Lijun Ni, Lauren Rich, Allison Elliott Tew, and Svetlana Yarosh, whose work I have re-interpreted in this article. Special thanks to Neil Brown, Brian Dorn, Barbara Ericson, and Tom McKlin for giving me comments on an early draft of this paper. I am grateful for the detailed and useful feedback from the anonymous reviewers. Thanks to Bill Leahy, Elijah Cameron, and Charles Isbell for their help in gathering updated data on the Georgia Tech MediaComp course. The framing of the paper grew out of discussions with Josh Tenenber, for which I'm grateful. This paper is based upon work supported by the National Science Foundation under grants #0618674 and #0634629 and under the *Georgia Computes!* BPC Alliance, Grants #0634629 and #0940394. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

9. REFERENCES

- [1] AAUW. *Tech-Savvy: Educating Girls in the New Computer Age*. American Association of University Women Education Foundation, 2000.
- [2] ACM and IEEE. Computing curriculum 2001. <http://www.acm.org/sigcse/cc2001>, 2001.
- [3] T. Balch, J. Summet, D. Blank, D. Kumar, M. Guzdial, K. O'Hara, D. Walker, M. Sweat, C. Gupta, S. Tansley, J. Jackson, M. Gupta, M. Muhammad, S. Prashad, N. Eilbert, and A. Gavin. Designing personal robots for education: Hardware, software, and curriculum. *IEEE Pervasive Computing*, 7(2):5–9, 2008.
- [4] J. Bennedsen and M. Caspersen. Failure rates in introductory programming. *ACM SIGCSE Bulletin*, 2:32–36, 39.
- [5] W. P. Dann, S. P. Cooper, and B. Ericson. *Exploring Wonderland: Java programming Alice and Media Computation*. Pearson, 2009.
- [6] B. Dorn. Scriptable: supporting informal learning with cases. In *Proceedings of the seventh International Workshop on Computing Education Research*, ICER '11, pages 69–76, New York, NY, USA, 2011. ACM.
- [7] B. Dorn and M. Guzdial. Graphic designers who program as informal cs learners. In S. Fincher, editor, *Proceedings of the International Computing Education Research Workshop*, New York, NY, 2006. ACM, ACM.
- [8] B. Dorn and M. Guzdial. Discovering computing: perspectives of web designers. In *Proceedings of the Sixth international workshop on Computing education research*, ICER '10, pages 23–30, New York, NY, USA, 2010. ACM.
- [9] B. Dorn and M. Guzdial. Learning on the job: characterizing the programming knowledge and learning strategies of web designers. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems (CHI 2010)*, New York, NY, 2010. ACM, ACM.
- [10] B. Dorn, A. Tew, and M. Guzdial. Introductory computing construct use in an end-user programming community. In *Proc. IEEE Symposium on Visual Languages and Human-Centric Computing VL/HCC 2007*, pages 27–32, 2007.
- [11] B. J. Dorn. *A case-based approach for supporting the informal computing education of end-user programmers*. PhD thesis, College of Computing, Georgia Institute of Technology, Atlanta, GA, 2012.
- [12] J. S. Eccles. Who am I and what am I going to do with my life? personal and collective identities as motivators of action. *Educational Psychologist*, 44(2):78, 2009.
- [13] A. Forte and M. Guzdial. Computers for communication, not calculation: media as a motivation and context for learning. In *Proc. 37th Annual Hawaii International Conference on System Sciences*, pages 10 pp.–, 2004.
- [14] A. Forte and M. Guzdial. Motivation and non-majors in computer science: Identifying discrete audiences for introductory courses. *IEEE Transactions on Education*, 48(2):248–253, 2005.
- [15] M. Furst, C. Isbell, and M. Guzdial. Threads: how to restructure a computer science curriculum for a flat world. In *SIGCSE '07: Proceedings of the 38th SIGCSE technical symposium on Computer science education*, pages 420–424, New York, NY, USA, 2007. ACM.
- [16] M. Guzdial. A CS1 course on media computation. <http://coweb.cc.gatech.edu/mediaComp-plan/uploads/1/working-document-v3.html>, May 2002.
- [17] M. Guzdial. A media computation course for non-majors. In *ITiCSE '03: Proceedings of the 8th annual conference on Innovation and technology in computer science education*, pages 104–108, New York, NY, USA, 2003. ACM Press.
- [18] M. Guzdial. *Introduction to Computing and Programming in Python: A Multimedia Approach*. Prentice-Hall, Upper Saddle River, NJ, 2004.
- [19] M. Guzdial. Education: Teaching computing to everyone. *Commun. ACM*, 52(5):31–33, 2009.
- [20] M. Guzdial. Does contextualized computing education help? *ACM Inroads*, 1(4):4–6, 2010.

- [21] M. Guzdial and B. Ericson. *Introduction to Computing and Programming in Java: A Multimedia Approach*. Prentice-Hall, 2005.
- [22] M. Guzdial, B. Ericson, T. McKlin, and S. Engelman. A statewide survey on computing education pathways and influences: factors in broadening participation in computing. In *Proceedings of the Ninth Annual International Conference on Computing Education Research (ICER'12)*, pages 143–150, New York, NY, 2012. ACM.
- [23] M. Guzdial, R. Fithian, A. Forte, and L. Rich. Report on pilot offering of CS1315 introduction to media computation with comparison to CS1321 and COE1361. <http://coweb.cc.gatech.edu/mediaComp-plan/uploads/45/FINAL-report-pilot-offering.1.pdf>, May 203.
- [24] M. Guzdial and A. Forte. Design process for a non-majors computing course. In *SIGCSE '05: Proceedings of the 36th SIGCSE technical symposium on Computer science education*, pages 361–365, New York, NY, USA, 2005. ACM Press.
- [25] M. Guzdial and E. Soloway. Teaching the nintendo generation to program. *Communications of the ACM*, 45(4):17–21, 2002.
- [26] M. Guzdial and E. Soloway. Computer science is more important than calculus: The challenge of living up to our potential. *Inroads – The SIGCSE Bulletin*, 35(2):5–8, 2003.
- [27] M. Guzdial and A. E. Tew. Imagineering inauthentic legitimate peripheral participation: an instructional design approach for motivating computing education. In *Proceedings of the second international workshop on Computing education research, ICER '06*, pages 51–58, New York, NY, USA, 2006. ACM.
- [28] R. R. Hake. Interactive-engagement vs traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses. *American Journal of Physics*, 66(1):64, 1998.
- [29] M. Hewner. *Student conceptions about the field of computer science*. PhD thesis, College of Computing, Georgia Institute of Technology, Atlanta, GA, October 2012.
- [30] M. Hewner and M. Guzdial. Attitudes about computing in postsecondary graduates. In *ICER '08: Proceeding of the fourth international workshop on Computing education research*, pages 71–78, New York, NY, USA, 2008. ACM.
- [31] J. Margolis and A. Fisher. *Unlocking the Clubhouse: Women in Computing*. MIT Press, 2002.
- [32] M. M. McGill. Learning to program with personal robots: Influences on student motivation. *ACM Transactions on Computing Education*, 12(1):4:1–4:32, Mar. 2012.
- [33] L. Ni. What makes CS teachers change?: factors influencing cs teachers' adoption of curriculum innovations. In *SIGCSE '09: Proceedings of the 40th ACM technical symposium on Computer science education*, pages 544–548, New York, NY, USA, 2009. ACM.
- [34] L. Porter and B. Simon. Retaining nearly one-third more majors with a trio of instructional best practices in CS1. In *Proceedings of the Special Interest Group on Computer Science Education (SIGCSE) Technical Symposium*, 2013.
- [35] L. Rich, H. Perry, and M. Guzdial. A CS1 course designed to address interests of women. In *Proceedings of the ACM SIGCSE Conference*, pages 190–194, 2004.
- [36] D. W. Shaffer and M. Resnick. “Thick” authenticity: New media and authentic learning. *Journal of Interactive Learning Research*, 10(2):192–215, 1990.
- [37] B. Simon, P. Kinnunen, L. Porter, and D. Zazkis. Experience report: CS1 for majors with media computation. In *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education, ITiCSE '10*, pages 214–218, New York, NY, USA, 2010. ACM.
- [38] R. H. Sloan and P. Troy. CS 0.5: a better approach to introductory computer science for majors. *ACM SIGCSE Bulletin*, 40:271–275, 2008.
- [39] E. Soloway, K. Ehrlich, J. Bonar, and J. Greenspan. What do novices know about programming? In A. Badre and B. Schneiderman, editors, *Directions in Human-Computer Interaction*, pages 87–122. Ablex Publishing, 1982.
- [40] J. Summet, D. Kumar, K. O'Hara, D. Walker, L. Ni, D. Blank, and T. Balch. Personalizing CS1 with robots. In *SIGCSE '09: Proceedings of the 40th ACM technical symposium on Computer science education*, pages 433–437, New York, NY, USA, 2009. ACM.
- [41] A. E. Tew. *Assessing fundamental introductory computing concept knowledge in a language independent manner*. PhD thesis, College of Computing, Georgia Institute of Technology, 2010.
- [42] A. E. Tew, B. Dorn, W. D. Leahy, Jr., and M. Guzdial. Context as support for learning computer organization. *J. Educ. Resour. Comput.*, 8(3):1–18, 2008.
- [43] A. E. Tew, C. Fowler, and M. Guzdial. Tracking an innovation in introductory CS education from a research university to a two-year college. In *Proceedings of the 36th SIGCSE technical symposium on Computer science education, SIGCSE '05*, pages 416–420, New York, NY, USA, 2005. ACM.
- [44] A. E. Tew, M. McCracken, and M. Guzdial. Impact of alternative introductory courses on programming concept understanding. In *Proceedings of the first International Computing Education Research Workshop*, New York, NY, 2005. ACM SIGCSE, ACM.
- [45] D. W. Valentine. CS educational research: a meta-analysis of SIGCSE technical symposium proceedings. In *Proceedings of the 35th SIGCSE technical symposium on Computer science education, SIGCSE '04*, pages 255–259, New York, NY, USA, 2004. ACM.
- [46] T. VanDeGrift. Encouraging creativity in introductory computer science programming assignments. In *Proceedings of the American Society for Engineering Education Conference*, 2007.
- [47] S. Yarosh and M. Guzdial. Narrating data structures: the role of context in CS2. In *ICER '07: Proceedings of the third international workshop on Computing education research*, pages 87–98, New York, NY, USA, 2007. ACM.