How Do Computing Faculty Adopt Curriculum Innovations? The Story from Instructors

Lijun Ni
School of Interactive Computing
Georgia Institute of Technology
Atlanta, GA, 30332-0760
lijun@cc.gatech.edu

Tom McKlin
The Findings Group
1031 North Hills Drive
Atlanta, GA, 30033
tom@thefindingsgroup.com

Mark Guzdial
School of Interactive Computing
Georgia Institute of Technology
Atlanta, GA, 30332-0760
guzdial@cc.gatech.edu

ABSTRACT
This paper presents the findings of an exploratory, qualitative study revealing computing instructors' experience in adopting curriculum innovations. We interviewed eight instructors a year after they attended workshops on several innovative introductory Computer Science (intro CS) courses at undergraduate level. The interview was designed to elicit the extent to which instructors had adopted or adapted what they learned from the workshops, and what drove or prevented their efforts to make curriculum change. The results of this study reveal that the adoption and adaptation of computing curriculum innovations in new situations may involve systemic change affecting instructors, departments and institutions as a whole. The findings of this study suggest a list of questions that a computing instructor might ask before committing to a new innovation. We also consider implications of this study for disseminating computing education innovations.

Categories and Subject Descriptors
K.3.2 [**Computers and Education**]: Computer and Information Science Education–*computer science education.*

General Terms
Design, Experimentation, Theory.

Keywords
Computing Instructors, Adoption, Change

1. INTRODUCTION
Due to the relative young and evolving nature of the computing discipline, the Computing Education community is faced with frequent changes to technologies they are teaching and teaching with. Innovative approaches to teach computing are being created to address the challenges we are facing in computing education (e.g., the evolving computing knowledge, the declining enrollment and interest in CS) [11] , as well as to improve the quality of computing education. These innovations include new pedagogical approaches, curriculum and educational software aimed at promoting learning and teaching in computing.

SIGCSE'10, March 10–13, 2010, Milwaukee, Wisconsin, USA.

To make those innovations or best practices have real impact on teaching practices, we eventually need computing instructors to adopt those innovations and integrate them into their own classrooms. Therefore, it is critical to understand how instructors actually adopt, adapt and implement computing education innovations in the contexts of their departments and courses.

Meanwhile, one major channel to disseminate educational innovations is through offering professional development (PD) opportunities for teachers [5, 9]. Related research indicates that faculty development participation has strong, statistically significant impact on faculty's adoption of innovative teaching practices [6]. Therefore, in particular, we are interested in exploring, how a computing instructor actually adopts (or fails to adopt) a curriculum innovation into the local school context after attending PD activities related to that innovation.

We interviewed eight CS faculty members from different universities about a year after they attended workshops on several innovative intro CS courses. This interview was designed to elicit the extent to which those instructors had adopted or adapted what they learned from the workshops. Furthermore, we attempted to understand what contributed to instructors' actual decisions on whether to make change within their local contexts.

In the following section, we introduce our methods, present the findings from this study, and discuss the implications of these findings for disseminating computing education innovations.

2. INTERVIEW STUDY
As Seidman [8] recommended, interviewing is a basic mode of inquiry into people's experience, which provides access to the context of people's behavior and thereby provides a way for researchers to understand the meaning of their action. In this study, we use interviewing as our method to start exploring computing instructors' curriculum change experience. We conducted semi-structured interviews with eight CS faculty members who had attended at least one summer workshop on several innovative intro CS courses for undergraduates.

2.1 Interview Population
The instructors attended at least one of the three workshops offered in summer 2007. These workshops introduced several contextualized computing courses offered for undergraduates to encourage diversity and to improve the enrollment and retention of students in CS [3, 4]. These courses emphasize the use of a specific context or theme (e.g. media manipulation or robotics) throughout the whole course, which students recognize as being authentic and relevant for computing [4].

The workshops were organized into 4-5 sections each day, around example lectures, follow-up exercises and discussions on how instructors could implement them into local school contexts. During the workshops, the workshop lecturer presented related research results, providing evidence that these contextualized curricula contributed to improvement on student motivation and retention [1, 10]. In order to help the instructors build confidence about using the contextualized approach, we encouraged them to work on learning tasks proposed in each course, e.g. creating an image collage computationally.

Interview participants were selected based on an adoption survey administered to collect participants' adoption decisions several months after the workshops [7]. From 24 instructors who reported adoption decisions, we randomly picked up eight instructors (four *adopters* and four *non-adopters),* who were willing to be interviewed. All of them were CS faculty members from different four-year universities, with experiences in teaching computing courses from 2 years to over 15 years. To improve the reliability of the interviews, we asked the second author to conduct the interviews, who was the external evaluator of this project.

2.2 Interview Questions

We used a semi-structured interview protocol to elicit the extent to which the instructors had adopted or adapted the contextualized computing courses and what were the influencing factors. The interview protocol was designed based on Guskey's [2] five levels of professional learning. Specifically, the participants were asked what content and pedagogical techniques they recalled from the workshop and how they applied those techniques. They were then asked what action they had taken as a result of their workshop participation, whether their participation resulted in any organizational change to their department, curriculum, instruction, etc. They were also asked to describe any student success they had observed related to using those innovations. Finally, they were asked about their future plans to adopt those courses. Throughout the interview, the participants were explicitly asked to reflect on what contributed to their actual adoption decisions.

2.3 Interview Analysis

Each interview was recorded and transcribed. We then performed a multistep analysis to increase coding reliability and fidelity. First, two authors examined four separate interview transcripts each to obtain an initial understanding of how computing instructors adopt curriculum innovations in their local school contexts. The individual transcripts were read several times. The portions of the transcripts directly related to instructors' change experience (e.g., adoption status, adoption factor, etc) were extracted for each transcript. Each researcher separately came up with themes or codes to categorize the extracts. The authors then came together to review the variations of computing instructors' adoption experiences. The authors then created a *code book*, a list of codes with a definition and an example of each code. Using this code book, one author then reviewed and coded all eight transcripts. During this process, categories of description were formed and reformed to capture the variation of the eight instructors' change experience. In the end, the authors reviewed and discussed the final coding results. The main results of this analysis are presented in next section.

3. RESULTS

In this section, we outline how the participants actually made or did not make any changes, based on what we found from the interviews. First, we summarize participants' reasons for considering curricular and pedagogical changes to teaching computing. We then describe their most pressing concern: making the innovation fit and work in a local department with multiple competing needs. We present the competing needs in terms of what prevents or facilitates instructors' change. Finally, we derive a set of questions from the interview transcripts that instructors contemplating a computing education innovation may consider.

3.1 Reasons for Considering Change

The major motivation for the instructors to look for curriculum innovations came from a need to change the current situation: lackluster student motivation, declining enrollment and retention of students in CS, and shrinking participation among women and under-represented minorities. The illustrative extracts below are what the faculty told us about what drove their intent to adopt the curriculum innovations from the workshops.

> P1: Enrollment really dropped. That is one reason why we tried to look at other ways and make changes in intro classes. The retention rate is also bad. We have a retention rate of like 30% in the school, and in our department, it was like 10%. There're serious problems. We had to think about what was wrong and the main problem was the intro class. So we tried to make it more interesting to get them to stay in the program.

> P3: The enrollment of women in CS has fallen off precipitously. That started 20 years ago, but now, it's a rather low percentage. I have about one woman in one of my senior classes. And that's not very many at all.

3.2 Adoption Status

While the faculty we interviewed were motivated to make changes and interested in adopting the curriculum innovations, all faced roadblocks to making the changes actually occur. The extent and nature of the roadblocks *determined* the status of their adoption.

- One participant can be described as a full adopter. He has adopted one Intro to Media Computation course *successfully* for two semesters, and then his department has decided to adopt it in the whole department in the coming semester.
- Another participant tried one Media Computation course in 2007. He has decided to teach it again this coming Fall when will be his turn to teach the CS1 course in his department.
- Another participant has chosen to adopt one element of the workshop, a new IDE, into his CS1 course. Now he and his colleagues who also teach that course are using this new IDE.
- One participant is hoping to adopt in the near future. She is waiting for a textbook for the new Media Computation Data Structures course to be available.
- The remaining four participants has expressed interest in adopting in the future; none has said that they has given up hope of adopting the innovation altogether.

3.3 Changing Organizations

The analysis of the interview transcripts also reveals that the adoption of computing curriculum innovations by individual

instructor may involve organizational change beyond one individual instructor. Here, the code "organizational change" is used to describe the process of getting members of the computing department to support or conduct an innovation. Most of the conversations on the topic of organizational change occurred among the non-adopters. Although many instructors described the flexibility they enjoyed in changing their own teaching strategies, a majority of these comments focused on the difficulty of getting one other person in the department to change in order to introduce both the content and teaching strategies inherent in the innovation.

> P8: There are four of us that are willing to try things and there is one of us that absolutely would not. I don't think you are going to get a whole departmental change to embrace Media Computation at this point.

> P7: I think it has possibilities for our CS0 course, but we haven't acted on that yet because the person in charge of that course did not leave. We were thinking of changing this if we took over that course or if we had responsibility for hiring somebody else.

While many claimed that the root of their adoption issues lied in getting one other person in the department to change, the real issue might not lie with that one person but with the organizational system supporting that one person's behavior. This coding category describes barriers that prevent the original adopter from introducing an innovation to a local computing department. Interestingly, this coding category was used more often than any other coding category.

3.4 Barriers to Change

The implementation of an innovation will face numerous barriers. These barriers can either slow down or completely stall a change that an individual instructor wants to create, despite the overwhelming need to try something different. Below, we outline some of the common barriers from the transcripts.

3.4.1 Little Freedom for Content Change

Many respondents said that they enjoyed the freedom to teach CS content in whatever way they wished, provided that the material was covered, the learning objectives were met, and the students were well-prepared for the next level. However, that freedom did not extend to changing what content was covered. Respondents felt that the new approach required a change to the content.

> P3: You know, the course content is already designed, and I think [the new] approach is very interesting. But, I don't have that flexibility to change the content. There are other faculty members teaching the course, so I have to use the course that is already developed.

3.4.2 Demand for Faculty Development

Adopting a curriculum innovation may require instructors to take time to learn something new, which competes with other faculty demands. In the following examples, the participants expressed the concern that adopting the innovation required training other instructors, a commitment that might over-burden stressed faculty.

> P7: Some other issues that are important are about faculty development. I have been to these workshops. I am familiar with the [innovations]… How are we going to do the faculty development? We made some changes before when moving to an object-oriented approach for our first course, and that was a lengthy transition.

> P4: The entire faculty is really overloaded with trying to meet teaching loads, advising loads and committee loads. So for the younger faculty members, that is going to be a real issue.

3.4.3 Poor Ability and Background of Students

Instructors might perceive that the change simply would not work within their own schools. For example, some participants felt their students' ability and background was a bad match with the innovation. They perceived that the students at the innovating institute might handle the innovation more easily.

> P5: This is too difficult for our kind of students. It might work well for students at [the university that developed these courses], but our students are a little different. Usually they are weak in foundations even if they went to CS0, CS1 and CS2. When they come into my class, some of them have a hard time to write even just simple programs or read programs. One really can't expect they already have any major math skills to learn programming. I don't think I would be able to incorporate that, at least for the data structures class.

3.4.4 Different Focuses on Learning Outcomes

New curricula may have different emphasis on learning outcomes. Faculty may disagree with this change. Some participants explicitly questioned the innovation because it strayed from a traditional approach and traditional values in teaching intro CS.

> P5: I think myself have some disagreement with the department that a lot of things that would go a little bit away from hardcore programming and still Media Computation requires a low level programming. There is a lot of programming in our program. Generally, some people want to push for less programming and others want to keep the status quo. So, there is disagreement in which way to go.

By changing the priority of learning outcomes, a new approach might attract students who prefer the new values. Faculty may not want those new students in their program. Another participant worried that a non-traditional approach might attract the *wrong* students and possibly set them up for failure later in the major.

> P8: I guess [Media Computation] is very visual, highly interactive, but there are other pieces to CS. There is a lot of it that is related to applied mathematics: computer theory, assembly language and machine organization, and all these kinds of components aren't going to be drawing pretty pictures and highly visual. In fact the [content] is going to be rigorous and very formal, requiring analytical skills that some people don't want to bother with. So I am not sure if it would attract the right people for the right reasons.

3.4.5 Near and Far Ripple Effect of Change

The participants also mentioned that change to first courses would result in cascading and possibly unexpected changes, both to the

courses in which the innovation was adopted and to courses later in the sequence. They reflected on what they currently valued in a course, and the affect of the innovation on subsequent courses.

> P6: The new course really requires rethinking of two things: 1) How do we teach fundamentally the same material? 2) What's important? What are the really important concepts here? If we change, it means that some of what we used to do, we no longer do, and some of what we are doing now we probably will be deemphasizing if we take the new approach. Is that good? Is that bad? It takes discussions and different faculty will react differently to this.

> P5: It does have an effect not only in the beginning courses. What can we assume that the students already know in subsequent courses? So there is a ripple effect throughout several courses in the curriculum.

3.5 Accelerators to Change

Just as there are barriers to adopting an innovation, we also see systemic conditions that not only motivate the innovation but might increase the rate at which that innovation may be adopted. We call the latter "accelerators."

3.5.1 Sense of Urgency

One accelerator was a strong sense of urgency about issues outlined in the section above, "Reasons for Considering Change." All participants were troubled by the declining enrollments in the department, low retention rates in CS, and homogenous student body. Some participants perceived this as a reason enough to try an innovation, whereas others might not be as affected by these trends and might be less likely to adopt. One participant summed it up by saying, "I think what we're finding here is that having a real traditional CS curriculum is not working."

3.5.2 Perceived Benefits for Students

The adopters among our interview group believed that motivating students was essential for increasing enrollment and retention, and the new approach would help to motivate their students.

> P4: I think students would enjoy Media Computation, and I think it would be valuable for them. It would help them understand computing a little bit, and it would be an interesting introduction for that purpose. Students seem to be more interested because they have a sense of ownership to the objects they are manipulating.

Faculty members were willing to invest efforts to change when they believe their change would help their students.

> P2: I really do believe students can learn this material and I do want them to learn it. And it's a way to bring more people into computing. Already, I've seen at least two people that have come over in our department and they have told me it's because of my class. So, I feel like I am making that type of positive impact. It's probably for their best interest and partly mine to make sure I am keen on those things that really help them.

3.5.3 Successful Experiment

One participant actually conducted an experiment comparing two courses, one using the new approach and the other taught in a traditional way. The *successful* initial adoption convinced him to further adopt this innovative course, even drove a full adoption in the whole department later on.

> P1: We did an experiment this year. I was teaching the course using Media Computation and my colleague was teaching the course in the way we've always been teaching it. We have discovered that the students seem to be responding better and they are more excited about the new course. The performance of students taking the new CS1 course in the next course (CS2) is on a par with the students who are coming out of the other courses. The plus side of all this is we've got more students finishing the new CS1 course successfully than in the traditional course. I think I've won over my colleagues and everybody in the department will be using Media Computation in the coming Fall.

3.5.4 Recommending Pedagogy

The participants mentioned earlier that they usually did not have the flexibility to change course content, which prevented their adoption. The courses presented at the workshops included change to pedagogy, suggesting particular ways of presenting CS concepts and new kind of student assignments. These were sometimes seen as a kind of *restrictions* for the faculty. However, some instructors perceived these *restrictions* as a positive step.

> P1: Traditionally, we've been able to teach what we want, just in the constraints of language that we're using. Now you are telling us how we are going to teach the course. And that is a shift in the way we have done things. But this was the right thing to do. When we switched from C++ to Java a couple years back, there was no discussion about how to teach the course, just "here is a different language to teach it". So the faculty members still pretty much designed their own approach to teaching. With [the new approach], there is a little bit more constraint. There is more focus on the types of assignments students will do even though there is latitude in picking the exact assignments.

Faculty might enjoy the freedom to choose the way to teach as long as the material is covered. However, under this situation, they may be inclined to focus more on covering the required material than on determining whether the students have learned the material. Otherwise, when an innovation brings corresponding pedagogy as well, faculty would likely implement a wholly change involving pedagogical innovations.

3.6 Critical Issues in Making Change

We derived a set of questions from the interview transcripts that a potential adopter may ask or consider before committing to an innovation. These questions were harvested from all interviews and reflected questions that were most asked by the faculty while considering adoption. These questions are:

- What is the important material or content for the course, and how will the innovation highlight the important content?

- What actual changes will I have to make to implement the innovation?
- How will we ensure that students learn the most important concepts and gain the same skills with the new approach?
- How might current and incoming students respond to the innovation?
- How will the innovation motivate students to enroll and persist in computing major?
- How will the innovation help us maintain the quality (of the computing course and thereby the quality of students)?
- How will the innovation assure us that students are well-prepared for upper-level computing courses?
- How will this change affect other stakeholders in our computing department?

Some stakeholders could be lab instructors, teaching assistants, instructors teaching the same course or upper-level courses, and instructors in other departments participating in a joined program. For example, a computing faculty might consider how the change might influence upper-level instructors: How will this curricular change prepare students for upper-level courses? Will upper-level instructors have to cover material that is not learned earlier?

These questions indicate different types and stages of adoption concerns from instructor who are considering and experiencing change: *personal concerns* about how change will affect themselves, *task concerns* about how to manage the new practice, and finally *impact concerns* about how the new practice will affect students and other stakeholders [5, 9]. In general, early concerns are more self-oriented, e.g. what is the innovation? How will it affect me? Our participants reported mainly the latter two concerns. One possible explanation might be: participating in related PD program helped to solve their early concerns. In other words, they had gained some knowledge about the innovation from the workshop, which helped to answer their early concerns.

4. DISCUSSION

This study provides an initial source of empirical data to explore how computing instructors adopt curriculum innovations. First, the results of this study indicate that one pressing concern for instructors (who want to adopt a curriculum innovation) is making the innovation fit and work in a local department. Adopting a new curriculum might become a major change involving systemic change affecting instructors, departments and sometimes other departments or even the institution as a whole. In addition to macro-level organizational change, successful adoption might also require micro-level change or agreement in terms of instructors' perceptions about the field and the learning goals to achieve. For example, in section 3.4.4, some interviewees *blamed* the new approach in distorting the image of CS. What underlies that disagreement is not just about different approaches, but more about different perceptions on what is CS and what is the most important to teach in intro CS courses.

Second, the findings of this study are consistent with an earlier study surveying CS teachers' adoption concerns in the end of the workshops [7]. Our results also indicate the complexity of implementing change in new contexts. Although the instructors we studied wanted to create change (had a strong motivation already), and had attended helpful PD activities, they still experienced a variety of challenges when making actual adoption.

Meanwhile, our findings are actually based on what the instructors believed as necessary to make change and the barriers they had encountered. It's fair to say that what instructors say are not necessarily the most critical issues for curriculum change. However, since instructors are the ones who actually implement the innovations in classrooms, it is critical to take into account their perceptions in order to effectively disseminate and implement curriculum innovations. For example, researchers and innovative curriculum developers might want to provide pedagogical recommendations and lecture examples within a new curriculum for easier adoption. Professional developers and facilitators need to understand what concerns PD participants face, and try to address those issues (example questions are listed in section 3.6). Particularly, when designing and organizing PD activities to disseminate innovations, PD developers and facilitators should consider how the PD could help motivate instructors to change and solve early stages of concerns [5].

5. ACKNOWLEDGMENTS

We would like to thank all the instructors participating in our study. This work is supported by the National Science Foundation under Grant Nos. 0618674 and 0634629. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

6. REFERENCES

[1] Forte, A. and Guzdial, M. Computers for communication, not calculation: media as a motivation and context for learning. in *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, IEEE Computer Society, Washington, 2004, Track 4, 40096.1.

[2] Guskey, T.R. and Sparks, D. *Evaluating Professional Development*. Corwin Press, 2000.

[3] Guzdial, M., & Forte, A. Design Process for a Non-majors Computing Course. in *Proceedings of SIGCSE 2005*, St. Louis, Missouri, USA, 2005, 361 - 365.

[4] Guzdial, M. and Tew, A.E. Imagineering inauthentic legitimate peripheral participation: an instructional design approach for motivating computing education. in *Proceedings of ICER 2006*, ACM Press, 2006, 51-58.

[5] Hall, G.E. and Hord, S.M. *Implementing Change: Patterns, Principles, and Potholes*. Allyn and Bacon, Boston, 2001.

[6] Matney, M. *Institutional and departmental factors influencing faculty adoption of innovative teaching practices*. School of Education, University of Michigan, 2001.

[7] Ni, L. What makes CS teachers change? Factors influencing CS teachers' adoption of curriculum innovations. in *Proceedings of SIGCSE 2009*, 544-548.

[8] Seidman, I. *Interviewing as Qualitative Research: A Guide for Researchers in Education and the Social Sciences*. Teachers College Press, 2005.

[9] Smith, C. and Gillespie, M. Research on professional development and teacher change: implications for adult basic education, NCSALL, 2007, 226-234.

[10] Tew, A.E., Fowler, C. and Guzdial, M. Tracking an innovation in introductory CS education from a research university to a two-year college in *Proceedings of SIGCSE 2005*. ACM Press, 2005, 416-420.

[11] Vegso, J. Interest in CS as a major drop among incoming freshmen. *Computing Research News*, *17* (3).