

STUDENT CONCEPTIONS ABOUT THE FIELD OF COMPUTER SCIENCE

A Thesis Proposal
Presented to
The Academic Faculty

by

Michael Hewner

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in
Human-Centered Computing

School of Interactive Computing
Georgia Institute of Technology
January 2011

STUDENT CONCEPTIONS ABOUT THE FIELD OF COMPUTER SCIENCE

Approved by:

Professor Mark Guzdial,
Committee Chair
School of Interactive Computing
Georgia Institute of Technology

Professor Amy Bruckman
School of Interactive Computing
Georgia Institute of Technology

Professor Keith Edwards
School of Interactive Computing
Georgia Institute of Technology

Professor Ellen Zegura
School of Computer Science
Georgia Institute of Technology

Professor Yasmin Kafai
School of Graduate Education
University of Pennsylvania

Date Approved: January 2011

TABLE OF CONTENTS

I	INTRODUCTION	1
1.1	Conceptions of CS	4
1.2	Why Conceptions of CS May Be a Problem	6
1.2.1	Conceptions are Persistent	7
1.2.2	Conceptions can Cause Educational Problems	7
1.2.3	Conceptions Vary Between Fields	8
1.2.4	Summary	9
1.3	What Understanding Conceptions Would Help Us Do	10
1.3.1	Advising	10
1.3.2	Students That Are Hard to Reach	11
1.4	Proposal: Studying Computer Science Conceptions	11
1.4.1	Thesis Statement	12
1.4.2	Research Questions	14
1.5	Summary	16
II	PREVIOUS WORK	17
2.1	Alternative Conceptions	18
2.1.1	Alternative Conceptions in Science Education	18
2.1.2	Alternative Conceptions in CS Education	21
2.2	Differing Expert Definitions of a Field	22
2.2.1	Differing Expert Definitions of Science	22
2.2.2	Differing Expert Definitions of Computer Science	23
2.3	Relationship Between Field Conceptions and Learning Content	25
2.3.1	Field Conceptions and Transfer	25
2.3.2	Field Conceptions as Representing Disciplinary Approaches	26
2.3.3	Field Conceptions as Guides for Learning	26
2.3.4	Summary	27

2.4	Studies of Student Conceptions	28
2.4.1	Studies of College Student Conceptions Outside of CS . . .	28
2.4.2	Studies of Student Conceptions of CS	31
2.4.3	Summary	36
2.5	Conceptions of CS in High School Students	37
2.6	Summary	41
III	STUDY DESIGN	45
3.1	Study 1: Eliciting Student Conceptions	46
3.1.1	The Choice of Grounded Theory	46
3.1.2	Data Sources	47
3.1.3	Interview Method	51
3.1.4	Analysis to Develop a Grounded Theory	52
3.1.5	Analysis of the Preliminary Survey Instrument	54
3.2	Study 2: Assessing Prevalence of CS Conceptions	55
3.2.1	Participants	55
3.2.2	Analysis	55
3.3	Summary	56
IV	PLAN FOR COMPLETION	58
APPENDIX A	STUDY 1 MATERIALS	59

CHAPTER I

INTRODUCTION

Talk to a student about what they are learning in an introductory programming course, and you might find a mismatch between what they are taught and the practical skills they hoped they learn:

After coming to Tech, I thought that the CS 1315 would really aid in my understanding of how computers worked. I was looking for a class that could teach about diagnosing common computer errors and the functions of each part within the system. However it seemed like just a novel way to write script using a programming system that I have not touched since. I instead learned about computer systems through CS majors that I made friends with. By my junior year I could replace the hard drive in my laptop, tell someone about the memory needed for certain systems, and understand the deeper inter-workings of the system.

—Biology major autobiography excerpt [32]

This sort of criticism is particularly poignant because it often comes from motivated students that are excited about what Computer Science has to offer. It’s a disappointment that the student felt that she learned nothing about “how computers worked.” But why was this connected to diagnosing common computer errors and replacing hard drives? This student came to class with a preconception about Computer Science that encompassed skills that an expert would not associate with the field. If the student had come to the course without preconceptions about what she would learn, is it possible she would have felt she did come away learning something

about how computers worked? Perhaps — but educators can’t assume that students come to our introductory classes as a blank slate.

Most CS educators have experience with students who come to CS with misconceptions about what the field of Computer Science is. Some examples that educators frequently mention anecdotally are:

- Students may come to CS expecting to learn “advanced features” of existing application software.
- Students may come to CS expecting to learn how to do IT work like assembling computers from parts and configuring routers.
- Students may come to CS thinking CS is just about programming and not understand why learning theory or architecture is valuable.

In departmental materials, textbooks, conversations with students, and occasionally in class, we attempt to educate students about what CS really is. No one in CS education research is sure exactly how common these sorts of misconceptions are. Educators hope that students in time come to an accurate view of what the field of Computer Science is, but understanding the field of CS is usually not a curricular goal or something we assess. Most importantly, the computer science education community does not know much about how this understanding of the field of Computer Science changes over time.

Of course not all views of Computer Science that differ from experts are the result of confusion. Students may understand the “expert” point of view and feel that view is wrong:

- Students may feel the credential of a CS degree is important, but feel that the actual material covered is not useful to them in the “real world”.
- Students may feel that understanding CS is important, but only as part of a larger different goal that is not purely CS (e.g. manager, computational artist).

- Students may have a position in the ongoing debates on what CS “ought to be”. They could feel that CS needs to be more human-centered, interdisciplinary, or theoretical, etc.

Some educators might wish to encourage at least some of these viewpoints; certainly there is value in a spirited debate about the direction of CS going forward. Regardless, similar to basic confusion, students who have different conceptions of CS may want different things from their CS classes.

When educators design their classes, obviously they take into account the fact that student goals and expectations may be different. But because little is known about how CS students conceive of the field, educators must rely on their intuition and ad-hoc conversations with students. This is made even more difficult when the effects of CS conceptions mixes with the ordinary confounds of college life. Feeling

like a class is unrelated to one’s goals can make a student work less, but so can laziness or scheduling conflicts. Discovering computer science was not about IT work can make a student leave the major, but so can frustration or a negative classroom culture. Our intuition is not enough to give a clear idea what student conceptions exist. If we knew what conceptions student have about CS as they go through their undergraduate careers, we could gear our curriculum and advisement to resolve confusion and accommodate variety of student conceptions.

In this document, I propose to study student conceptions of the field of Computer Science. My approach will be cognitive: I will study what students think CS is, how that affects decisions students make, and what causes conceptions of CS to change. Based on related research, I argue that conceptions about the field of CS take time to change, conceptions affect student learning and educational decisions, and CS conceptions are different from what we know about student conceptions of other disciplines. I argue that we can use a more thorough understanding of conceptions of CS to better guide our student advisement and teaching. I propose to study student

conceptions of CS through analysis of materials we use to teach about the field of CS and interviews with advisors and students. From these interviews, I hope to develop a grounded theory that can be used to build a preliminary survey to elicit student conceptions of CS on a broader scale.

In this chapter, I present an overview of my approach, the arguments that understanding conceptions of CS are important, and my overall plan with the research contributions. In chapter 2, I provide discussion of related work in the general areas of science education and the computer science education community in particular. In chapter 3, I provide details of my method including elicitation techniques and the type of analysis I intend to do.

1.1 Conceptions of CS

There are many ways we could examine student perspectives on CS. Other research has considered a variety of factors: if students think CS is fun or a good match for them [30], if students are unduly affected by stereotypes [33], if the culture of CS is welcoming to new participants [44, 52], and many other aspects. I focus on the cognitive aspect of CS major’s perspective of the discipline; I am interested in what majors think CS is and what they expect to learn by studying CS.

Much of my approach comes from the large literature of alternative conceptions in science [66]. In science education literature, an ‘alternative conception’ is simply an existing student understanding of science concepts that usually differs from the ‘expert-like’ understanding of their teachers. Alternative conceptions are usually about science content, not conceptions of science fields (e.g. they generally study if students understand evolution, not if students understand what Biologists do). Nonetheless, the alternative conception literature has established a variety of robust findings about students learning difficult concepts that have been applied to many areas of education and many groups of students.

My work differs from scientific alternative conception because there is no single correct view of the field of CS. Experts in CS have different views of the field [39] so it is reasonable to think students might have different but valid views as well. From an educational perspective, establishing a single universal definition of CS is not important. Students simply need a clear enough understanding of CS to choose good classes, attend to the right material in class, and make other ordinary educational decisions. Therefore, my definition of CS conceptions focuses on student understanding of the curriculum and educational choices.

I define a student's *conception* of CS to be what skills and concepts a student expects to learn in their CS curriculum and how they expect to use those skills and concepts after graduation. This includes an understanding of the reasons behind the curriculum: why particular concepts are covered and others are left out. Although there is expert disagreement about a formal definition of CS [39], the curriculum itself is fairly standardized [56]. Even if not all CS experts agree with the curriculum, it is reasonable to expect that students who do not understand the CS curriculum may make poor educational decisions.

This research classifies student conceptions into two categories: *productive conceptions* and *potentially problematic conceptions*. Productive conceptions are those that show a good overall understanding of how the parts of CS fit into a coherent whole and have a logical vision for how those concepts and skills will help after graduation. A student may have a productive conception even if they are not certain how some specific concepts (e.g. finite automata) fit into the overall picture. Overall, a productive conception is one that lets a student reason about CS and make informed educational decisions.

Potentially problematic conceptions are conceptions that in some way significantly deviate from the curriculum. A potentially problematic conception might be an outright misconception about CS (e.g. CS is training in using advanced features of

applications). A potentially problematic conception could also be simply overfocusing on some real aspect of CS (e.g. CS is programming). Of course, it is possible that a student with a potentially problematic conception of CS may never have a problem; the conception may be corrected before the student makes any significant educational decisions. A potentially problematic conception is simply a conception that has the potential to cause poor educational decisions.

What we do know about students concepts about CS suggests that, at least initially, many students may have potentially problematic conceptions. Similar to science concepts, we have evidence that few high-school students in the general population understand much about CS. Carter [8] reports that of students in a high school calculus and precalculus class, 80% of students reported that they did not know what Computer Science is. Qualitative research suggests similar results for a variety of age groups [43, 68]. My own research (described in more detail in Chapter 2), similarly supports the idea that even high school students who have been exposed to some Computer Science education still have significant misconceptions about the field of Computer Science. Also consistent with research on scientific conceptions [59], what evidence we have for older students suggests that views of the field of CS do change by the time of graduation [5, 68, 32].

1.2 Why Conceptions of CS May Be a Problem

CS education provides some evidence that many students come to CS with potentially problematic conceptions. What is not clear is if these potentially problematic conceptions actually cause poor educational decisions. I argue that the evidence we have suggests that potentially problematic conceptions are likely to persist longer than we expect, that they are likely to lead students to make poor educational choices, and that the types of potentially problematic conceptions are likely to be different in CS than they are in other fields.

1.2.1 Conceptions are Persistent

Taking a page from the large literature of science alternative conception research, we should be skeptical about assuming that students quickly overcome their initial potentially problematic conceptions about Computer Science. One of the most commonly observed aspects of alternative conceptions in the science literature is that teachers habitually underestimate how difficult it will be to change student conceptions [66]. Often, even after alternative conceptions are explicitly discredited in class, students will either misinterpret this new information as supporting the conception they still have or adopt parallel concepts: one for answering teacher questions, and one for reasoning in everyday life [27].

Even if students do change their conceptions after participating in CS classes, there is no reason to assume that the new concept of Computer Science is more expert-like than the old. Stevens [61] discusses how students in engineering often view “what makes a good engineer” in the light of their current coursework. In the beginning, student viewpoints in engineering are dominated by the view of an engineer as someone who works on highly constrained mathematical problems with known answers. When the coursework finally begins to be more open-ended and deal with real world problems, students confident in the “math problem” approach now have significant stress. There is evidence to suggest that this tendency to narrowly define the field by introductory coursework occurs in CS too. McGuffee [46] reports that while initially students’ definitions of CS are too broad, after one CS class students narrowly define CS as just programming.

1.2.2 Conceptions can Cause Educational Problems

Even if a student thinks Computer Science is simply training in how to use applications, they can still learn CS content like variables and loops. But we know from general psychology research that what information an observer is looking for greatly

affects what they notice [25]. Similar to research with experts and novices [15], understanding what features of information are important is an large part of the skills that differentiate experts and novices. If a student expects to learn about using applications, it is reasonable to suspect they might focus on how to use the IDE more than they focus on how to construct good algorithms.

Beyond simply doing well in classwork, students have to make educational decisions that rely on their knowledge of Computer Science. Students have to make decisions about what and how to study, whether or not they are well suited for the major they are on, what electives and extracurricular activities are worthwhile. All these things rely on how they conceptualize CS.

Finally, if majors begin to feel that teacher conceptions are out of line with what they perceive as reality, majors can put their focus elsewhere. Nespor [48] documents how, in one school’s management department, the majority of the majors believed that what they learned in class was irrelevant. Students developed their own culture: short term strategies for getting grades and fanatical focus on grooming and interview techniques (what the students felt companies were really looking for).

1.2.3 Conceptions Vary Between Fields

Every field probably has issues with majors not initially understanding the field. But although the fact that students have potentially problematic conceptions is similar, the ways in which each field develops student understanding is deeply involved with the individual concepts of the field. Because educational practices differ between fields, it is reasonable to expect that student’s conceptions change in different ways.

Studies in various disciplines definitely provide evidence that student development is different across disciplines. In Nespor’s studies [48], physics students and management students experienced very different trajectories as they advanced their majors, and these are different than the stages of development in engineering disciplines that

Stevens notes. Each of these disciplines has to deal with some similar challenges: students in each field feel that the actual professional practice of the discipline is distinct from the problems of classroom. In engineering, Stevens [61] notes that, as the curriculum progressed, professors moved more in line with “real-world” problems and constraints, as well as increasing training in laboratory skills. This caused significant stress as students dealt with having to apply more varied skills to less constrained problems. In Nespor’s physics students, by contrast, the essential problems remained similar but became more abstract: students might work a problem they had worked in previous semester but this time from more basic principles. Because physics is not a design-oriented discipline, physics students focused exclusively on the difficulty of theory and problem solving, and not vague requirements or managing large scale group projects.

There is also reason to think that Computer Science is a particularly interesting field to study conceptions. CS is a new field and even among experts there are a variety of opinions about what is and is not CS [39]. Probably even within individual departments, many potential viewpoints on what CS is represented. How do students make sense of this? It is even likely to think that students may hold opinions about Computer Science that are not common among experts today but will be as the field develops. In research in CS conceptions, it is especially important to keep in mind that while some student may have conceptions of CS that hurt them academically (and this is problematic), there are many potential conceptions of CS that may be different but perfectly productive for our purposes.

1.2.4 Summary

In the previous section we outlined several reasons to think that the intuitive viewpoint we have about CS conceptions is not detailed enough. From other research into alternative conceptions, we find evidence that alternative conceptions are resistant

to education and persist longer than instructors suspect. From general research in psychology and studies of other educational settings, we find evidence that student conceptions are likely to affect learning and behavior. From educational research, we find evidence that although conceptions exist in every field, they manifest in different ways and therefore it worthwhile to look at CS specifically.

1.3 What Understanding Conceptions Would Help Us Do

Of course, even if Computer Science has a unique set of persistent potentially problematic conceptions, that doesn't necessarily mean that understanding them will allow us to fix them. As the research in education shows, it is often easier to identify potentially problematic conceptions than it is to develop effective curricula to resolve them. But in the case of conceptions about the field of Computer Science, I argue that an understanding of student conceptions would provide several benefits.

1.3.1 Advising

The first thing that a detailed understanding of student conceptions would let us do is better advise students who have potentially problematic conceptions. Some students may enter Computer Science expecting training in 3D modeling or other non-CS goals: these students should be identified as quickly as possible and directed to the appropriate major. Attrition when students have picked the wrong major is unavoidable, but we don't know how many of our students leave for this reason. If we could tell how common basic CS misunderstanding is, then we could estimate how much of our attrition is caused by factors we can improve: frustration, negative learning environments, etc.

Understanding conceptions is useful beyond students who are in the wrong major. Some students may have productive conceptions of Computer Science, but yet find their goals out of sync with their professors or the prescribed curriculum. If this is true, we may need to think about how to support these students appropriately.

Current CS curricula is excellent for producing a certain kinds of skill, but if many students have goals in a different direction that is something to be considered seriously.

1.3.2 Students That Are Hard to Reach

Research into potentially problematic conceptions is particularly relevant if we want to advise students who are doing poorly in the Computer Science major. Different conceptions aren't necessarily equally likely among all students: we expect to see more potentially problematic conceptions among struggling students. This follows from the general psychological result that individuals are motivated to maintain a positive identity [47]. If one is doing well in classes, then obviously it enhances self image to perceive instructors as objective and in touch with the reality of Computer Science. For students who are doing poorly, perceiving instructors as arbitrary and out of touch with the reality of CS is a simple way to preserve an image of themselves as competent. This is natural and important: obviously not every student can be the top of the class and students should find areas where they can be considered exceptional. But a student's feeling of disconnection from the field can also allow them to excuse undesirable behaviors like cheating, as in the management majors Nespor studied [48]. Disconnected students may also have little motivation for going beyond what is required in class. If we want to encourage these students to participate in Computer Science, the first step is to understand how they view Computer Science.

1.4 Proposal: Studying Computer Science Conceptions

In the previous sections, I argued that:

1. There is good reason to think that students have different conceptions about Computer Science across their undergraduate career.
2. These conceptions are likely to affect their education in a variety of ways

3. If we understood how student conceptions are learned, it would be a good first step towards improved student advising and support

In this section I propose a research method to study student conceptions of Computer Science.

1.4.1 Thesis Statement

By examining the evidence of student concepts of Computer Science in undergraduate students, I believe it will be possible to:

1. Characterize changes in student understanding of the CS field
2. Develop a preliminary instrument that can be used to elicit student understanding of the CS field

RQ1: What types of CS field conceptions exist in CS undergraduate students?	H1. CS majors will exhibit a changing understanding of CS, initially potentially problematic but becoming more productive.	Grounded Theory analysis based on interviews with students, student surveys, advisors, and CS materials like textbooks and departmental materials	Study 1 n = ~25
	H2. Multiple productive conceptions will be found in graduating undergraduate students.	Grounded Theory analysis based on interviews with students, student surveys	Study 1 n = ~25
RQ2: Do potentially problematic CS conceptions effect student educational decisions?	H3. Potentially problematic conceptions of CS will effect educational decisions.	Grounded Theory analysis based on interviews with students	Study 1 n = ~25
RQ3: What is the prevalence of different kinds of conceptions among the CS major population?	H4. Students conceptions will vary across the student population, with potentially problematic conceptions persisting after introductory coursework.	Open ended survey instrument based on grounded theory	Study 2 n = ~60

1.4.2 Research Questions

RQ1: What types of CS field conceptions exist in CS undergraduate students?

To answer this questions I propose a qualitative study based on several data sources:

1. *Interviews with CS counselors.* Those who advise CS students have a large opportunity to examine a variety of student conceptions about CS, and especially how these conceptions factor in to students' educational decisions.
2. *Written materials.* CS departments and instructors often attempt to educate students about the field of CS. Departmental materials and introductory textbooks are likely places to look when seeing what the "official" view of CS is as presented to students.
3. *Interviews with students.* I plan to interview CS majors at two different schools about how they view CS and how they feel that view has changed over time.
4. *Survey Instruments.* As I work with students through interviews, I plan to refine an open-ended survey instrument with questions that elicit student conceptions of CS.

Departmental materials, student advisors, and students will be drawn from two different 4-year CS degree programs. Students will be from various stages in their major, and varying levels of academic success. To select which students to study, I will engage in theoretical sampling [62]: selecting which types of students would be most likely to provide new insights based on the analysis of interviews thus far. I will use Grounded Theory [62] to analyze all the data sources; the goal of the analysis will be to extract a general theory of the various student conceptions of CS at different points in the curriculum. In accordance with Grounded Theory practices, I will interview until the theoretical categories achieve saturation (i.e. new interviews do not generate new theoretical constructs) but I anticipate this will be approximately 25 interviews.

I hypothesize that students in the early stages of their CS education will reveal potentially problematic conceptions of CS, but that these conceptions will become more productive as students progress in the CS degree. Specifically, I expect to see students that anticipate they will learn non-CS content in their CS degrees. I also expect to see students that are unable to explain why what they are learning is considered important.

I hypothesize that students later in their CS curriculum will still have a variety of conceptions about CS. I predict that at least some students will be able to characterize their professors conception of CS, yet hold a differing viewpoint. I predict that these different conceptions will cause even students with productive conceptions to make different educational choices.

RQ2: Do potentially problematic CS conceptions effect student educational decisions?

I hope to answer research question 2 with the same qualitative study I proposed for research question one. I hypothesize that interviews with students will reveal connections between conceptions of CS and educational choices. Specifically, I predict that some questions in the interview will reveal that students with potentially problematic conceptions make different choices in what material is important to study, what extracurricular activities improve CS-related skills, and what courses to take. I also predict that students later in their educational careers will be able to identify mistakes they made because of their old conceptions of CS.

RQ3: What is the prevalence of different kinds of conceptions among the CS major population?

From the qualitative work in the previous study, I hope to devise a survey instrument that can be used to assess student alternative conception prevalence on a larger

scale. After developing the instrument, I hope to use to evaluate the prevalence of alternative conceptions among students at two different stages of the CS curriculum. Although the survey questions will likely be open-ended, the results will be evaluated by a rubric based on the conceptions identified in the previous part. For this part of the research, I plan to use inter-rater reliability metrics to verify that the instrument and rubric measures alternative conceptions in a systematic way. I also plan to attempt to correlate conceptions with other measures of academic success.

I hypothesize that identified potentially problematic conceptions will be present in many students. Specifically, I think that differences in conceptions will be clear between introductory students and experienced students, but that potentially problematic conceptions will exist past the first CS courses. This will suggest that conceptions do not quickly become productive and can affect educational choices in a long-term way.

1.5 *Summary*

This chapter has presented an overview of the argument that CS conceptions affect student thinking and are worth studying. Chapter Two presents related work about alternative conceptions in general, and what we already know about student conceptions about CS specifically. Chapter Three presents a more detailed description of the two studies proposed here, including a detailed description of the methods of analysis. Chapter Four presents a provisional timeline for completing my thesis work.

CHAPTER II

PREVIOUS WORK

This chapter covers five main topic areas:

1. *Alternative Conceptions.* My research uses the idea of alternative conceptions as the basis for analyzing student conceptions of CS. This topic summarizes the characteristics of alternative conceptions and discusses how they are normally studied.
2. *Differing Expert Definitions of a Field.* My work is made more complicated by the fact that there is no single “right” concept of Computer Science. This section discusses the controversy and how decisions of definition have real consequences for students.
3. *Relationship Between Field Conceptions and Learning Content.* This section reviews research that suggests that poor field conceptions can make it more difficult to learn content.
4. *Similar Studies of Student Development.* Though my focus on the cognitive understanding of the field of CS has not been examined specifically before, many studies have interesting results concerning how students conceptualize their field.
5. *Conceptions of CS in High School Students.* The one of the main inspirations for this proposed work is a study I did with high school students. This section discusses the method I used on that project and what I observed of high school student conceptions.

These sections draw on two different bodies of literature: educational research from other fields and CS-specific education research. Disciplines like science and engineering education research are often older and have established methods. CS-specific research give more information about the specific attitudes and problems that CS students have.

2.1 Alternative Conceptions

2.1.1 Alternative Conceptions in Science Education

Based on constructivism and the learning experiments of Piaget, alternative conceptions research begins with the idea that students often develop cognitive models of science that differ from experts. The “alternative conceptions” that students have are accurate enough for everyday life, and maybe even accurate enough to pass simple testing, but cause students to reason incorrectly in key ways. For example, a physics-based alternative conception is Aristotelian motion in which objects in motion naturally stop unless force is applied. This can explain most ordinary phenomena students come into contact with, but is very different from the way physicists view motion. By understanding how alternative conceptions affect student thinking, science educators believe that better educational approaches can be developed.

Science researchers have found students to have alternative conceptions in every area of science and these alternative conceptions cut across normal social boundaries like race, age, and cultural background [66]. But what is most interesting about alternative conceptions is that they are very often resistant to instruction; Wandersee et al. call this resistance “the most reported finding in the field” ([66] 190). Gilbert et al. [28] have classified the diverse ways students with an alternative conceptions may respond to instruction that contradicts their beliefs:

- Students may simply leave their original conception unchanged.
- Students may construct two views: one for answering instructor questions and

one for explaining real world situations.

- Students may misinterpret the new information as confirming their existing belief.
- Students may construct a hybrid idea with elements of the contradictory viewpoints.

These sort of responses underscore the need for careful analysis to determine if students have alternative conceptions. Poorly constructed instruments can mistake “hybrid ideas” or “two views” responses for an expert-like understanding. Given that, a large challenge in alternative conception research is designing reliable tools to determine what student alternative conceptions are.

2.1.1.1 How Alternative Conceptions are Elicited

The process of determining student alternative conceptions generally begins with a combination of instructor intuition and open-ended questions. One example is an early study by Osborne and Gilbert [50] in which students are asked general questions based on simple drawings. For example, students are shown a drawing of a man pushing a motionless car. The interviewer might ask if there is a force acting on the car. This single question can elicit a variety of alternative conceptions. A student may argue that there is no force because there is no motion (force implies motion). A student may argue that there is a force, because the person is “forcing” the car (confusion with everyday word “force”). A student may argue that there is a force, but that the car “wants” to stand still so it has no effect (ascribing desires to objects). All these are potential alternative conceptions that Osborne and Gilbert identify.

The main difficulty of initial alternative conceptions research is choosing good questions that elicit alternative conceptions. The man pushing car scenario is a good question because it highlights the differences between the students’ everyday

conception and that of experts. As physics educators, Osborne and Gilbert have a good intuition about the sorts of questions that make good starting points. Similarly, in other areas ([13] and [24]) educators use known problematic situations to form the basis for open ended questioning. Because science educators have a good intuitive understanding of what situations students will find difficult, initial research can start with a focused set of questions that students will have problems with.

In my proposed research, there is no clear intuition about what questions will elicit alternative conceptions about the field of CS. For alternative conceptions about science concepts, instructors routinely test student understanding formally and informally and know what areas to probe. For research on alternative conceptions on the field of CS, instructors have less experience asking good questions. For this reason I have chosen a more qualitative approach than most of the previous work: we need to explore broadly before we have an idea of the right questions to ask. Once we have the result of the qualitative interviews, I hope to develop a more structured testing tool that asks specific questions that will illuminate student alternative conceptions easily.

2.1.1.2 How Alternative Conceptions Drive Pedagogy

The purpose of identifying alternative conceptions is not simply to understand what “errors” students are making. Even if a instructor explicitly says an alternative conception is wrong in the classroom, students will often maintain their old viewpoints [59]. Many creative classroom techniques have been applied, some have proven successful in some cases, but no generally reliable method exists for encouraging students to change alternative conceptions to expert-like conceptions [66].

The benefit to understanding alternative conceptions is not simply to explain why students are wrong, but also to understand the ways in which students are right. Smith, DiSessa, and Roschelle [59] argue that alternative conceptions are established

because they work well for the practitioner and have many of the aspects of expert-like reasoning. The goal of appropriate constructivist pedagogy is to use students' existing understanding to construct a better point of view: existing alternative conceptions provide the resources necessary to do that [59].

One consequence of the need to work within the framework of existing alternative conceptions is that not every expert-like conception is equally valuable to learners. Lynn and Muilenburg [41] describes the development of heat curriculum that omitted the expert idea of heat being caused by atomic motion. Lynn and Muilenburg argue that this model does not relate well to student everyday experiences and that instead discussing "heat flow" provides a clearer basis for students to see the benefits of scientific conceptions of heat. Though the idea of students adopting expert conceptions is appealing, the goal is really to allow students to reason accurately and provide a good basis for learning in the future.

The research on the relationship between alternative conceptions and pedagogy has several implications for my research. One is simply that there is not necessarily a simple relationship between understanding alternative conceptions and developing pedagogy to make them expert-like. Also, this work suggests that CS educators need to build on the alternative conceptions students have—not simply attempt to replace them with the expert version.

2.1.2 Alternative Conceptions in CS Education

The idea of alternative conceptions has also been used before in understanding student conceptions of CS content. Even when not explicitly mentioning alternative conceptions, a great deal of CS education research focuses on student conceptions of programming, and how that understanding differs from experts (see [11] for an overview). Recently, work is being done that explicitly references the science education alternative conception community directly (e.g. [65, 31, 35]). This work has had

success in documenting alternative conceptions and designing instruments that can test for specific conceptions.

Although alternative conceptions have been used to understand problems students have with CS *content*, no work in CS has been done that explicitly uses the same approach to understand problems student have with the *field* of CS. Researchers studying student views of the field (discussed in detail later in this chapter) have focused either on student affect or on the social environment of CS. The alternative conception research I propose allows the CS community to look at views of the field of CS from a cognitive perspective, and understand how different conceptions of CS can affect student educational choices.

2.2 Differing Expert Definitions of a Field

Defining a field like CS is difficult. Both science and CS have a history of differing expert definitions, and these differing definitions do have educational implications. In this section I discuss the various definitions and describe how I will evaluate student responses despite the reality of differing expert viewpoints on CS.

2.2.1 Differing Expert Definitions of Science

Educators often agree that science education is important, but disagree about why. In *Rising Above The Gathering Storm*, the Committee on Science, Engineering, and Public Policy argues that science education is essential because innovation in science and technology are necessary for improved national well-being and competitiveness [16]. Other advocates argue that science is important because voting citizens must weigh in on scientific debates or because science teaches important habits of mind [57]. Each of these viewpoints on the goal of science education suggests different scientific curricula.

There are also those who argue that standard educational approaches to science education force students to adopt a westernized culture of science [1]. While this

might not seem like a bad thing to all educators, practically speaking, students who are unwilling or unable to adopt the rules of science culture can be alienated and not learn [1]. Moreover, if educators are serious in desiring a diversity of opinions in the science community, then the goal is definitely not to repress real objections that science is not moving in the right direction. It is a careful balancing act to teach students science content while at the same time not implying the scientists naturally know best or that certain types of people are not well-suited for science [4].

My research directly touches on these issues in regards to what is an appropriate definition for CS and what is not. As CS educators, we wish to encourage thoughtful debate on what CS ought to be and respect a variety of opinions. On the other hand, when students have a misconception that has the potential to harm them educationally, we hope educators can guide them to a better definition of CS. The controversy about science makes it clear that deciding whether a definition of is ‘good enough’ is not a value free choice: the goal is to allow a variety of valid conceptions of CS while understanding how some conceptions can cause educational problems.

2.2.2 Differing Expert Definitions of Computer Science

Debates about CS education parallel those of science education in many ways. There are those that argue CS teaches good habits of thought [51, 67], those that argue understanding CS is essential to good participation in modern society [54], and those who argue that CS is training for jobs that are in high demand [19]. Even some of the most famous practitioners of CS cannot agree if CS should be considered science, mathematics, engineering, or art [17]. Some of this disagreement is a matter of emphasis [39], but there are also arguments about whether subfields should be considered part of CS (e.g. [23, 45]). Even when formal definitions for CS have been attempted, they are not of a form that beginners could reasonably be expected to articulate or reason with (see [18] for one such definition). As CS mingles with other

disciplines, the distinctions become even more complex; Rosenbloom [53] presents an algebraic notation to describe the various ways CS can be combined with other sciences. This problem has resurfaced again with the interest in “computational thinking” : even among experts, a definition that everyone can agree with is elusive [14].

This controversy makes it clear that my research cannot compare student conceptions to a single expert viewpoint on Computer Science. But not all the distinctions that matter to experts have educational consequences for undergraduates: students can think of CS as primarily artistic or primarily mathematical, and still have clear views of what classes and content will help them achieve their goals. Because this research focuses on conceptions of CS that have educational consequences, what is most important is how students conceptualize the content of the undergraduate curriculum.

For all the discussion of ‘what CS is,’ the curriculum of CS undergraduate degrees has a standardized set of topics to cover [9]. This standardization is what I will use to try and assess student understanding. If a student has an understanding of the topics covered in CS and why they are considered important, I will consider that a productive conception of CS. If a student expects to learn things in CS that are not related to the curriculum (or, in retrospect, believes instructors intended him or her to learn something outside the curriculum), my research will consider that a potentially problematic conception of CS.

Of course, it is also possible for students to understand the reasons they are taught something, yet feel that some other content is actually more useful to them (e.g. understanding that data structures was the point of the course, but feeling that the C++ programming language is actually more useful). Although the reasons behind the student’s choice are definitely worth pursuing in that case, by the definition of this research, the student conception of CS is accurate. Given the wide ranges of goals and viewpoints in CS, no one can second-guess what educational choices are

correct for an individual student. Students must be arbiters of what they wish to learn; but it is important they make decisions with a clear understanding of CS.

2.3 Relationship Between Field Conceptions and Learning Content

It is clear that conceptions of CS can directly affect certain educational choices: students with problematic conceptions of CS may major in CS when another field (e.g. informatics, graphic arts) is much more closely related to what the student envisions. What is less clear is if potentially problematic conceptions of CS can influence learning directly; is a student with a potentially problematic conception of CS less able to learn ordinary CS content like algorithms? There is some research that suggests that having a problematic conception of the broader field can directly hurt students' ability to learn content.

2.3.1 Field Conceptions and Transfer

One simple connection between an understanding of the overall field and learning content is the issue of transfer. Whatever CS students' career goals are, they will have to take a variety of courses that on the surface seem abstract and distantly related to the skills students hope to develop. One of the robust findings of the literature of learning transfer is that students often fail to connect abstract ideas with their applied context unless they see the connection when these ideas are taught [6]. This is one of the factors that makes teaching for transfer frequently unsuccessful [2]. If students do not have a conception of CS that explains why CS concepts are important, it is possible that they can learn the concepts in their original classroom context and then be unable to transfer the ideas to later courses.

2.3.2 Field Conceptions as Representing Disciplinary Approaches

Another connection between understanding of the overall field and learning content is the idea that there are implicit ideas at the “field level” that underlie approaches to content. Donald argues that each discipline has its own essential methods of approaching problems that are distinctive and difficult for students to internalize [20]. For Donald, a significant part of education within disciplines is conveying these ways of approaching problems. A student expecting to use everyday (or even rigorously scientific) approaches in a discipline where these approaches are not valued is sure to meet with failure.

Deanna Kuhn expresses an example of this idea of discipline-wide thinking with her analysis of argumentation and experimentation [37]. Kuhn’s study asked students and adults to reason about social problems (e.g. why prisoners commit crimes after release) as well as concrete experiments (e.g. determining what factors made simulated cars go faster). In both cases, with both children and adults, Kuhn found her subjects unable to differentiate between evidence and a plausible story to justify assertions. Even factors like expertise in the issue under consideration did not improve the quality of argumentation. However scientists were able to correctly design experiments [36] without trouble. Kuhn argues this suggests that scientific reasoning is not a simple extension of common sense. Although the design and analysis of experiments may never be explicitly taught, reasoning correctly about experiments represents a conceptual approach that people do not naturally develop without instruction.

2.3.3 Field Conceptions as Guides for Learning

A third connection between understanding of the overall field and learning content is the idea that an accurate view of the field allows learners to view themselves as more capable. This idea is analogous to Dweck’s work on intelligence [21], who showed that learners who viewed intelligence as something that came from their own effort

were more academically successful. Songer et. al [60] attempted to establish a similar connection for science: that students who viewed science as facts to be memorized would do more poorly than students who viewed science as integrated knowledge that could be understood through reasoning. In the study, students who viewed science as facts to be learned rather than as explanations that could be changed had greater difficulties learning thermodynamics. Although this was just one study, it does provide some evidence that attitudes about how a field like science ought to be learned does affect learning outcomes.

2.3.4 Summary

There is no simple well supported connection between field conceptions and success in learning content. However, all three of these lines of research suggest plausible ways that student conceptions of CS might affect learning of CS content:

- *Field Conceptions and Transfer.* Anecdotally, students do frequently complain to professors that they do not understand how courses like computer architecture and theory are related to their goals. Students who do not see why these subfields are useful may well find it difficult to transfer what they learn into practice.
- *Field Conceptions as Representing Disciplinary Approaches.* Because disciplinary approaches can be implicit in the thinking of experts, it is difficult to know if CS has discipline-wide approaches that students struggle with. This will be something to examine in my interviews, however.
- *Field Conceptions as Guides for Learning.* We know that people outside of CS often view programs as mysterious and fixed tools rather than as designed artifacts [55]. In CS courses, teaching that programming languages and hardware could have been designed differently is often a goal. There is no concrete evidence that viewing technology as mutable improves student success but similar

work in science does suggest it is possible.

Overall, there is evidence that understanding field conceptions may influence content learning. The number of studies that attempt to connect field conceptions and content learning is small; these studies are suggestive but do not establish a clear link. I hope to explore this relationship further in my work and attempt to see if there are clear ways that conceptions of CS affect learning CS content.

2.4 Studies of Student Conceptions

2.4.1 Studies of College Student Conceptions Outside of CS

Several studies have attempted to understand how students relate to their college major. Certainly students do not always make a well researched choice of major, and students' major choice are influenced by factors beyond how much interest and aptitude they think they have [34]. Students often change major in college, and the major change is often to a related field [3] which may suggest that they refine the partly uninformed choice they made when they arrived.

Several studies have focused on engineering students. Engineering has similar problems to Computer Science in that it has high student attrition [22]. Looking at the problem both qualitatively and quantitatively, Edward claims that students initially envision engineering as hands on with a mathematical component, but are unprepared for the level of abstraction in the introductory curriculum. As their career in engineering continues, students find the later courses more practical. In the end, 66% of the students felt that their classes prepared them well and looked forward to their career while “the remainder were dreading what they saw as a continuation of slogging through mathematical calculations” [22]. Edward recommends that the curricula explicitly connect between the theory and everyday engineer practice in order to prevent attrition and disenchantment with the major.

In a study that reports similar findings, Stevens [61] analyzes the results of longitudinal interviews with students in engineering programs. Stevens emphasizes that processes which seem homogeneous student recruitment, development, and eventual success or failure in the major result in idiosyncratic student experiences. Students attempt to display what Stevens calls “accountable disciplinary knowledge” (ADK): things that, in the view of experts, position them on the path to engineering. Students reason about the nature of the field, using the ADK they are expected to display as a guide, but what counts as ADK varies depending on the point in the curriculum and other contextual factors. As a result of this, when the ADK students are expected to display changes they experience stress. Students who initially struggle with the ADK of introductory courses can distinguish themselves later. Failure to display the expected ADK causes students to be shut out of the major or makes the students question their appropriateness for engineering. Stevens argues that deep qualitative analysis is necessary to discover the way in which the structures of the curriculum influence different kinds of development.

ADK is one way of thinking about the effect of a distant, disembodied field on the education of individual students. The field (which includes both industrial and academic experts) shapes what counts as ADK. ADK then structures the experience of students in the major. But simple departmental requirements are only part of the picture: students communicate with each other and develop a intuitive understanding of what the requirements mean [61]. Similar to the social groups discussed in Lave and Wenger [38], students form a community of practice that transmits an understanding of overall field structure. But unlike the groups studied by Lave and Wenger, the community of practice of engineering students at particular school is not really independent. If students are to come to an accurate view of their field, their community of practice must be somehow connected to larger distributed community of practice of engineers. How does a local community of practice relate to this larger

discipline? This idea of a distant community of practice is examined in Nespor's [48] study of physics and management students. For both physics and management students, the community of other students transmitted a very distinctive conception of their field.

Nespor's [48] observation of the practices of physics students suggest that the students and teachers use objects like textbooks to replicate a consistent community of practice. Nespor argues that physics's highly abstract notation and traditional problems makes this replication possible. Physics content shaped the community of students; it encouraged them to devote a huge amount of effort to constantly solve textbook problems. Because the act of solving these traditional problems takes so much effort, physics students need to work together in groups that make a community of practice locally at the school. Because the work has as its focus textbooks, lectures, and the actions of a few teachers, the community is consistent with physics student communities across many schools. The consistency by which problems and notations can generate this similar community of practice is what makes physics a discipline that can replicate itself across long distances (though, as Nespor points out, social and structural factors are in place to make this replication possible).

Nespor's example of management majors provides an interesting contrast. In this case the curriculum is disjointed and did not build to generalized abstractions. Students generally did not have a respect for what they were learning in class; they viewed their instructors as out of touch with the true practice of management. But the majors nonetheless adopted a set of practices based around appearance and interpersonal interview-type skills. In this sense, the implicit curriculum management majors adopted mimicked some of the characteristics of the physics curriculum (consistent, requiring practice, involving personal investment and elaboration outside of class). Nespor argues that this might even have been correct, that perhaps management majors were right in their view that the true management community of

practice was based around the interpersonal skills they practiced. It is definitely true that they reproduced the practices of a real community beyond the individual college. In both management and physics, student effort and the content of the field allowed students to feel they were a part of a community that they were not directly connected to.

This research suggests several things for my research. Firstly, because CS, like engineering, is not easily understandable by introductory students, students may struggle to understand what counts as ADK in CS. It may be reasonable to suspect a variety of idiosyncratic paths through the CS major.

Nespor’s research on how communities of practice are replicated suggests a deep relationship between the actual content of the discipline and the ways students come to learn it. Certain structures within CS may encourage students to focus on the academic curriculum, as in physics, others may encourage students to construct their view of disciplinary knowledge in contrast to their instructors, as in management. Whatever the result, it seems clear that careful qualitative analysis of student concepts of CS is likely to be able to draw a connection between students understanding of CS and the way they make educational choices.

2.4.2 Studies of Student Conceptions of CS

2.4.2.1 Research on Conceptions of Precollege Students

Much of the research in conceptions of the field of CS has focused on the perspective of precollege students, generally middle school and high school students. It is difficult to characterize the results simply: students are often positive about technology, but simply enjoying doing “CS type” activities does not translate into feelings that CS is a good career [42]. Students obviously use technology regularly, but often have concerns about a computing career being “sitting in front of a computer all day” [68]. Developing accurate instruments is difficult [30] because small changes can significantly alter how students respond. The large-scale WGBH study [26] of students age

13-17 indicates that careers in Computing interest students across ethnicity, although young men like computing more than young women.

It seems that students do not have a ready definition for Computer Science in general. Greening [29] asked high school students to complete the sentence “Computer Science is mostly about...”: the majority said they didn’t know or provided trivial answers like “computers”. Only a small percentage mentioned using applications, something that others have identified as a student misconception. Similarly, in a student of high school calculus students, Carter [8] found that 80% of students left blank the question “What is your impression of what Computer Science Majors learn? (leave blank if you have no idea)”. Of course, the general population may not be a good guide for the attitudes of those who major in CS: obviously everyone who chooses CS as a major as at least some preconceptions for what it will be about.

2.4.2.2 Research on the Postsecondary Community of CS

At the postsecondary level, several qualitative studies have been done to characterize the student experience in the CS major. The most well-known of these is Margolis and Fisher’s study of women in the CS major at Carnegie Mellon university [44]. Margolis and Fisher focus on barriers to women’s full participation in CS, and their work has many interesting reflections on how female students contrast with their male counterparts. But the work also has several tantalizing hints that reflect both the diversity of student conceptions entering CS and their evolution over time. Students entering CMU view CS as a natural fit with their role as computer wizards, an opportunity to pursue larger social goals, or as part of a opportunity to achieve financial success. As time progresses for students who stay in the program, a growing connection to CS both on a personal and academic level develops ([44] pg. 103–107). Margolis and Fisher look at this problem through a social lens: we hear about students’ feelings of greater integration and success, but less about cognitive changes. Does the diversity

of initial views of CS eventually converge, remain fixed, or change to another set of diverse viewpoints? Margolis and Fisher provide an excellent illustration of the deep description a qualitative study can provide, but there are many things that remain to be understood about the student experience in the CS major.

Another suggestive study is Rasmussen and Håpnes's analysis of three social groups within the CS major: computer hackers, dedicated students, and "normal" students. Based on interviews with students, the authors argue that the three groups view both the social environment of the major and CS in a different way. Most interesting, Rasmussen and Håpnes argue that the perspective of the majority of students (the group the authors term as "normal" students) is actually defined in contrast to that of the professors. To the "normal" students, the professors were disconnected from the practical reality of computing jobs and too similar to the hacker subgroup that the normals did not wish to associate with. In terms of the nature of CS, "normal" students were interested in user interfaces and other parts of CS that were considered practical. How these views were reconciled into an overall concept of CS is again outside of the scope of the primarily social analysis that Rasmussen and Håpnes focus on.

The qualitative research that has been done highlights the stressful, occasionally unsuccessful, ways in which students are integrated in CS. Many of these barriers are social, but both of the studies above also highlight the fact that students vary in their reasons for pursuing CS and that these reasons change as the students learn. Obviously, it is important that departments ease cultural problems that make students feel like they do not belong. But both these studies suggest that CS educators need to understand the various perspectives on the field, and expect that these alternative conceptions of CS will have real consequences in what students expect in the curricula.

2.4.2.3 Research on How CS Student Conceptions Change

While more is known about student conceptions at the beginning of the CS major, there is evidence to suggest that the student conceptions do change. McGuffee [46] describes student responses to the question “What is Computer Science?” He reports that at the beginning of CS1, student conceptions are too broad, while at the beginning of CS2 students definitions are too narrowly focused on programming. This is consistent with Steven’s research [61] that students overfocus on what they are taught in introductory classes.

Biggers et al. [5] compares conceptions of CS in seniors: some of whom left the CS major and some of whom stayed in the major to completion. One of the main differences between the stayers and the leavers is that stayers are more likely to define CS broadly while leavers were more likely to define CS as simply programming. Because the majority of CS students left earlier in their careers, there are two possible interpretations of this result. One is that all students initially think of CS as programming, but then that conception changes as they are exposed to more courses. The other is that students with broader conceptions are likely to persist in the major, while those who think of CS as just programming are likely to leave. Either way, the study suggests important relationships between changing conceptions and student retention.

A third study on conceptions focuses on student conceptions about software engineering. Sudol and Jaspan measured student agreement with statements about software engineering that were tested on experts to ensure correctness [63]. They found that students had misconceptions compared to experts, and in general these misconceptions decreased over time. However, project courses in operating systems and web applications seemed to increase misconceptions, despite the fact that these courses are taught by faculty with real development experience and focus on software engineering concepts. The authors hypothesize that the group work in these classes is

not realistic enough and therefore causes students to endorse bad practices. Clearly, even when students have been exposed to expert viewpoints, they readily develop potentially problematic conceptions based on their own observations and experiences.

2.4.2.4 Attitudes About Computing in Postsecondary Graduates

A previous study of mine also provides evidence of CS conception change for CS majors near graduation. The goal of the study was to identify whether student's college experiences, including interest-targeted introductory CS courses, had an affect on student attitudes about computing four years later. This project compared essays from students in different majors about computing; one of the interesting results of this study was how different CS majors' essays about computing differed from the other majors.

One of the difficult parts of this project was eliciting student experiences in a way that was not leading. Previous work had shown that students were positive about their CS courses [64]; the goal of the study was in part of to see if students would bring those courses up unprompted as significant computing related experiences. The method we chose was based on the techniques of Schulte and Knobelsdorf [55] who asked beginning CS majors and 3rd year psychology students to write "computing autobiographies". These autobiographies revealed interesting differences in attitudes, while giving participants a great deal of control to discuss whatever they felt was personally significant.

In my study, we collected data from 4th year students at Georgia Tech [32]. The autobiographies of CS majors were clearly different from students in other majors. CS majors concentrated their autobiographies on the breadth on the discipline. Other majors were often extremely enthusiastic, but focused on technology itself as a fun thing to play with. If the student population can be considered similar to those interviewed by Schulte and Knobelsdorf [55], it is reasonable to suspect that this focus

on the broad possibilities of Computer Science represents a change in conception from CS freshmen. It is also consistent with the results of Biggers [5] and Yardi [68] which similarly find CS majors who focus on the breadth of the discipline. From this, I suspect it is likely that this breadth-focused CS conception may be at least one of the conceptions my proposed work is able to elaborate. Given the limited data in the autobiographies, it is difficult to know if this breadth-focused conception leads to good educational choices.

Although this study provides some hints for what may result, the autobiography format has several disadvantages. First, though it allows many students to respond, because they are fully in control of what is written about, there is no way to probe understanding. The focus of the analysis therefore, must be on what students chose to mention rather than the cognitive aspects of their conceptions. Second, the ability to analyze a large number of autobiographies is less useful in qualitative work: it is more useful at this stage to analyze deeply and sample students who's attributes will contribute most to the researcher's tentative understanding [62]. Still, as one of the few studies to probe CS conceptions for students later in the major, it is important foundation for my hypothesis that students conceptions change throughout a students time in a major.

2.4.3 Summary

Previous research into student field conceptions has several interesting results. From studies in engineering and other fields, I expect students to have a evolving conception of their field that is different from what their professors might envision. From the work in pre-college conceptions of CS, I expect students to enter the CS major with at least some confusion about the field. From the work in later college conceptions of CS, I expect to see a broad view of CS in at least some students. But although all these studies are suggestive, overall the CS education community has a vague picture

of how students think about CS as they spend time in the major.

2.5 Conceptions of CS in High School Students

My most recent research is a study of high school students' conceptions of CS. This study provided clear evidence that students, at least before formal training in CS, have large misconceptions and these misconceptions can be difficult to dislodge. The group high school that was studied was usual: all the students who had experienced a great variety of interesting CS interventions: fun approaches designed to focus both on programming and the innovative potential of CS [7], but most of the students had never taken a 'formal' CS class.

The study method consisted of four parts:

1. *Concept Map Instruction.* We gave the students one hour introduction to concept maps, based on instructions in [49]. Finally, after the hour long introduction the students were given twenty minutes to build a concept map about Computer Science.
2. *Pre-Interviews.* After the students build the concept maps, individual students were called into interview. The interview focused initially on the student's concept map and asked them to explain their reasoning, with particular attention to areas that suggested misconceptions. Then the interviewer presented a few example activities and asked if these activities are part of Computer Science and if so where they fit on the student's concept maps. Finally, the interviewer asked some questions how the student might interest a friend in Computer Science and what attributes the participant consider essential for success in Computer Science.
3. *Class.* A week after the interview, the students attended two 4-hour class sessions (1 week apart) that attempted to further elaborate their concepts of

Computer Science.

4. *Post-Interviews.* A week after the last class session, the students were asked to build a second concept map about Computer Science. They were given a copy of their original concept map to use as a reference if they wished. After building the map they were interviewed following a similar script to the pre-interviews. As part of the post-interview, the students were asked to compare and contrast their concept map of four weeks ago with their current one.

Both from the concept maps and interviews, it was clear that students had significant potentially problematic conceptions about CS. For example, students frequently believed that CS could train students for non-CS jobs that simply involved computers. For example:

Interviewer: [What would you say if someone asked] “If I were to get a degree in Computer Science, could it be my job to use photoshop professionally?”

P5, pre: I’m sure. I mean Pixar and all the Disney companies they are using digital art media now. All their movies are digital pretty much. Marketing too there’s a lot of digital applications to design marketing advertising sorts of things.

As in the alternative conception literature [66], student conceptions colored their perceptions. Students frequently remarked about a presentation they had received about how CS could be used in medicine. However in the students’ interpretation, CS people were responsible for using computers in places like hospitals.

Futhermore, there seemed to be ways in which student conceptions of CS influenced their educational choices. One of the students we interviewed was considering enrolling in Computer Science at Georgia Tech as part of the Digital Media specialization of CS. Contrasting this to a traditional art degree, he/she said:

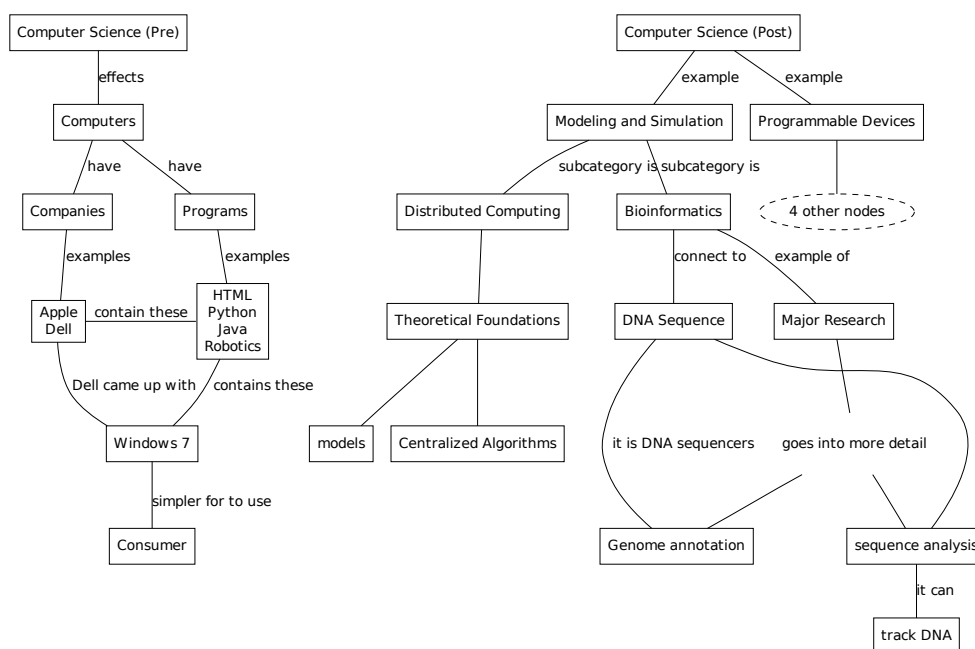


Figure 1: Pre and post concept maps about Computer Science (digitized from the handwritten form for clarity). You can see both the student has a more recognizable categories within CS, but that within these categories there is still confusion.

P4, pre: You could probably get more into graphics and creating art with the computer and animating things. Where as in the art [program] you might be more dealing with the pure painting sketching sculpting.

Though certainly one could use the understanding of computer graphics and similar topics in the CS degree to create innovative art, this student seems to be envisioning something more similar to a digital media program at an art school.

What made this situation even more concerning was that students did not, in general, discard their potentially conceptions after they were addressed in class. One problem was that there are few resources for understanding the real internal structure of the field of CS in a way beginning students can understand, outside of an expert explaining. Contrasting before and after concept maps (see Figure 1) the student's conception of CS has more recognizable categories, but within the categories there is

still confusion. Even for concepts students can readily understand conceptually, like distributed algorithms, the resources available outside of the classroom are so focused on experts that students can't understand what they discover on their own:

Interviewer: So let's look at your modeling and simulation concept categories. First off, you have under there distributed computing. What is it?

P3, post: It's kind of like um when I researched it it was kind of like finding stuff kind of I looked on a couple I think I looked on one website and learned a little about it I thought it was like technical part of computers kinda...

Interviewer: So you said theoretical foundations and under that you put centralized algorithms and models. So explain what's a theoretical foundation and why are centralized algorithms and models beneath that?

P3, post: Well I didn't really know that much about theoretical foundations. So when I kinda went under that these were the categories that were underneath it. I really don't know what a centralized algorithm is but I wanted to take some more time to learn that...it looked interesting.

It is from this study that I take my some of my key hypotheses:

- Students may have potentially problematic conceptions about CS.
- These potentially problematic conceptions influence their educational choices.
- These potentially problematic conceptions are resistant to change.

There are obviously significant differences between the students in this preliminary study and the groups I have proposed to study. These students (in general) did not intend to study CS professionally. They are high-school students and therefore are

in a very different environment that CS majors are. It is possible that in a college environment and with formal CS courses, they would come to a good understanding of CS before any significant educational problems occurred. But this study does suggest that conceptions are potentially something that can cause unexpected problems and is a worthwhile area to study.

2.6 *Summary*

This chapter reviewed related work in five main areas. Here is a summary of the most important points to take away:

1. *Alternative Conceptions.*

Alternative conception research is a large area of study, especially in the sciences. The basic premise is that students come to the classroom with existing ideas about how the world works called alternative conceptions. These conceptions are resistant to change, especially to classroom instruction techniques that simply present the ‘expert’ viewpoint [66]. Understanding alternative conceptions is important not just because they must be addressed, but because these existing conceptions provide the resources from which students necessarily construct new knowledge [59].

My work uses the idea of alternative conceptions in a new way. This work focuses on student conceptions of a field (i.e. CS), rather than individual concepts within the field. It is reasonable to think understanding a field is similar to understanding other difficult concepts. However, because educators do not have strong understanding of student conceptions about CS, qualitative analysis must happen before educators can develop the tests traditionally used to assess student conceptions.

2. *Differing Expert Definitions of a Field.*

Both science and CS must deal with the fact that there is no single expert definition of the field that everyone can agree on [57, 17]. In both science and CS, there are concerns that certain definitions make it more difficult for some students to participate [4, 44]. This creates a difficult situation: educators wish to encourage a diversity of opinions while at the same time correcting conceptions that are problematical.

The controversy makes it clear that this research cannot simply compare student conceptions to a single ‘expert’ view. The goal of this work is to focus on conceptions that cause students to make poor educational choices: therefore if students understand the basic curriculum of CS [56] and why it is considered important then their conceptions are likely accurate enough to prevent educational mistakes.

3. *Relationship Between Field Conceptions and Learning Content.*

There are several lines of research that link field conceptions and learning. The educational literature of transfer suggests that understanding the reasoning for learning concepts is important in applying them beyond the classroom [2]. Donald suggests that individual disciplines have implicit approaches that students need to understand [20]. Songer and Linn connect field conceptions to individual learning strategies [60]. All of these research approaches have studies that connect student field conceptions to learning outcomes in specific instances.

The diversity of these results makes it difficult to predict how much conceptions affect learning outcomes. Each of these studies link specific conceptions to specific academic performances: it is not clear whether any of these processes will be visible at the large granularity of student interviews. The research that exists does support the idea that field-level conceptions influence the way students solve certain kinds of problems. In that sense, it does support my

argument that student conceptions of CS can affect learning in subtle ways.

4. *Studies of Student Conceptions.*

In engineering and science, qualitative studies have shown that students often perceive their majors in unexpected ways. Stevens’s work [61] suggests that context strongly influences what skills students associate with the major, and therefore how successful they perceive themselves to be. Nespor argues that the structure of the disciplinary knowledge itself creates different social structures that change student experience. Both authors show the benefits of detailed qualitative analysis to uncover how academic structures affect student learning in unexpected ways.

In CS education research, research on precollege students suggests that students do not (in general) have a good definition for what CS is. Qualitative studies on the community of CS majors suggests that students do have different CS conception, and these conceptions cause them to make different educational choices [44, 52]. Survey-based studies provide evidence that student conceptions do change [46, 5]. But although all these studies provide interesting hints, no study has attempted to understand how student conceptions in CS change across all four years of their undergraduate career.

Although none of these studies examines student conceptions in the cognitive way I propose, many of them provide strong hints that student conception of CS are likely to be interesting. The qualitative approaches make it clear that student opinions are complex. The quantitative approaches provide evidence that even simple measures of student concepts of CS can reveal interesting statistical results. Hopefully the research I propose which uses qualitative analysis to help build a instrument to elicit alternative conceptions, will give us a much clearer understanding of the way students conceive of CS.

5. *Conceptions of CS in High School Students.*

Similar to previous work, my study of high school students provided evidence that high school students have poor conceptions of CS. Even when problematic aspects of students' were address directly in class, students maintained problematic conceptions of CS. At least in some cases these conceptions seemed to influence student educational decisions. Though this study was not explicitly designed with the idea of field-level conceptions, it provides a starting point for my proposed work. The methods I will use will be similar with a broader and more rigorous qualitative analysis.

In this chapter I have reviewed research related to alternative conceptions, both in education in general as well as within CS specifically. This research is what underlies my hypothesis that a cognitive view of field-level conceptions is a fruitful perspective that is likely to generate new understanding of the way students progress in CS. In the next chapter, I will focus on method: both the research that underlies the method I chose and my specific plans for studying student conceptions of CS.

CHAPTER III

STUDY DESIGN

In the previous chapters, I have argued that CS educators have a limited idea of what conceptions students have about the field of CS. My research attempts to advance this understanding by answering three research questions in two studies:

- RQ1: What types of CS field conceptions exist in CS undergraduate students?[Study 1]
- RQ2: Do potentially problematic CS conceptions effect student educational decisions?[Study 1]
- RQ3: What is the prevalence of different kinds of conceptions among the CS major population?[Study 2]

Both studies focus on the understanding of undergraduate CS majors. Undergraduate CS majors were chosen because a great deal of research has already been conducted on the views of high school students. By looking at undergraduate students, I hope to address the issues of retention and success in CS for students that already have some interest and familiarity with Computer Science.

The first study is an open-ended qualitative study designed to understand what conceptions exist and how they affect student educational decisions. The primary data for this study comes from interviews with undergraduate CS majors. The student interviews are supplemented with written sources of information about CS as well as interviews with student advisors. The data sources will be analyzed using Grounded Theory methods with the goal of producing an accurate understanding of the CS different conceptions students have.

Once a theory of student CS conceptions is developed, it is reasonable to try to assess student conceptions on a larger scale using something like a survey. As part of the first study, I propose to develop a open-ended survey instrument. As the study progresses, I hope to change the survey with the developing theory. My goal for the instrument will be to develop something that can reliably elicit similar information to the interviews.

The second study is a larger study of students in several undergraduate CS classes. I use the survey instrument to assess conceptions across a large groups of students. The results are evaluated with a rubric based on the theory built from the interviews. From this, I hope to determine how common different conceptions of students are, for several groups of students.

3.1 Study 1: Eliciting Student Conceptions

This section begins with a discussion of why Grounded Theory was chosen for this project. Then the various sources of data are discussed, including details of participants and recruitment. The section concludes with specific issues about interviewing and the analysis of interview of survey data. Appendix A contains the initial interview guide and initial survey (both of which are subject to change as the study progresses).

3.1.1 The Choice of Grounded Theory

I argue in Chapter 2 that because CS educators do not have experience eliciting student conceptions, an qualitative approach is necessary. Many different qualitative approaches have been used in education [58]. Some qualitative approaches are not suitable for interviewing a variety of students about CS (e.g. ethnographic observation or case studies); I believe interviews are the easiest and best ways to understand how students conceptualize CS. Within the framework of student interviews, however, there are still a several potential approaches.

Grounded Theory has several advantages:

- *Emphasis on developing a theory grounded in the participants.* One of differences between grounded theory and other approaches is its requirement that the theory emerge from the participants. Though (as discussed in Chapter 2) the CS education community does have some ideas about likely conceptions, there is also a high likelihood that the way students think about CS will be different from what we expect. Grounded theory gives the researcher the freedom to pursue what concepts naturally arise; to understand students conceptions of CS we need to be prepared to approach the issue on the students' terms.
- *Emphasis on integrating multiple data sources.* For this study, resources like departmental materials and CS advisors have the potential to provide key insights. Grounded Theory encourages multiple data sources as input to the overall theory.
- *Theoretical Sampling.* Because we do not know much about student conceptions, it is difficult to know what factors might influence student conceptions. As a result, it is premature to plan exactly what student groups to interview. This is part of the usual process of Grounded Theory: analysis occurs after each interview and the researcher selects their next subject based on the developing theory. This allows the researcher to fully explore interesting aspects of the data rather than attempting to achieve a random sample or vary certain predetermined variables.

3.1.2 Data Sources

In accordance with Grounded Theory [62], this study will work with several different data sources in its analysis of student conceptions of CS.

3.1.2.1 Written Materials.

I plan to look at several written sources as possible influences on student conceptions of CS. I plan to look at definitions of CS as expressed in:

1. Introductory textbooks: Students may or may not actually read textbooks and, as per alternative conceptions research, may not actually interpret what they read correctly [66]. But even if students do not read the textbook directly, the approach of the textbook writer is likely to be implicit in the design of introductory CS courses. I will focus on the sections that define Computer Science and if the overall structure of the book supports that stated definition.
2. Departmental materials: This category includes the departmental website, promotional materials, and the description of the major in the course catalog.
3. Major courses and requirements: The relationship between courses and other curriculum features and CS is both explicit and implied. The course catalog has explicit descriptions of courses and why they are important. But students are also likely to infer what is important in the field by looking at curriculum requirements including tough non-major courses or departmental application processes [61].

For all these materials, I will specifically focus on the textbooks and departmental materials of the two schools I will be interviewing students from.

3.1.2.2 Interviews with CS Counselors.

People in the CS department who advise students are likely to already have some understanding of common CS conceptions students have and how those conceptions influence student educational decisions. At Georgia Tech, students must speak to a counselor before switching majors from CS, so the counselors likely have experience with conceptions that cause educational problems. I hope to use these interviews to

understand what conceptions counselors notice in students and what sort of questions they ask to understand how students think about CS. I suspect that counselors will have their own theories about how student conceptions of CS change. Even if those theories are not all supported by my interviews and other data, some of them may suggest likely areas to probe in my interviews with students. Beyond that, I am interested to see if counselors find it easy to change potentially problematic conceptions students have or if students stick to their original conceptions despite counselors' advice.

3.1.2.3 Interviews with Students

I plan to interview students about their conceptions of CS and how those conceptions have changed over time. In accordance with theoretical sampling, I do not have a particular plan of exactly how many students I will interview or what groups each individual person will be drawn from. The interview process will complete when new interviews do not seem to add new concepts (see the discussion of saturation below). That said, the realities of research make some planning necessary. There are three categories of student that warrant special consideration:

1. *Time in Major.* One of the main hypotheses of this research is that conceptions change over time. I plan to interview students who are still within their first or second semesters in CS, students who are in their fourth or fifth semester, and students who are near graduation.
2. *School.* Because the populations schools draw from is different, and because curricula does vary between schools, I plan to interview students from two different schools: Georgia Tech and Spelman College. Spelman is a historically black college for women, and as such is demographically different from Georgia Tech on a variety of dimensions. Both schools are four year programs with curricula in alignment with the standard [9]. This is an important category that

needs to be planned for in advance because of the need for advance approval at both schools being studied.

3. *Level of Academic Success.* Because one of the goals of this study is to connect conceptions with academic success, and because there is reason to think that level of academic success might affect conceptions, I plan to look at students with varying levels of success in their classes. Because of the difficulty of working with long-term student data like transcripts, I plan to use instructor-evaluated student performance in class as a proxy for overall academic success. This is an important category that needs to be planned for in advance because of the care that must be taken whenever researchers obtain student grade information.

These three categories are not necessarily the only important factors in student conceptions of CS; they may not even become relevant as the grounded theory develops. Time in major is included because of its relationships with the research questions of this proposal. School and academic success are important because of research requirements that they be specified up front.

Recruitment will be done through presentations in CS classes. Students will be asked to volunteer and will be offered a gift certificate to compensate them for participating. Given that most of the classes will be large and only a few students will be interviewed from each class, finding enough students to interview should not be difficult. Students will be selected from classes based on the three criteria above and any other consideration that emerges from the interviews as the grounded theory develops.

3.1.2.4 Preliminary Survey Instrument

Students will be asked, as part of the interview process, to fill out a preliminary version of the survey instrument being developed for the second study. The main goal of this survey will be to validate that student conceptions of CS as elicited in

interviews are consistent with results reported in the survey instrument. The survey instrument will change as understanding of what questions get most clearly to likely student conceptions improves. For the initial version of the survey instrument, please consult Appendix A.

One of the components of the initial survey instrument is the use of a Computer Science concept map. This is based on the research by Novak and Gowin [49], that identifies a concept map as an excellent way to both allows students freedom in expressing complicated conceptions but also producing an artifact that is easy to evaluate for conceptual flaws. My previous work found the concept map to be extremely useful, showing both potentially problematic and productive aspects of individual students' conceptions.

3.1.3 Interview Method

At a high level, the goals of the student interviews are:

1. Determine a students conception of CS, how the student came to the conception, and if the conception is potentially problematic or productive.
2. Determine if a student feels their conception has changed, and if so how and why.
3. Determine how a student conception is influencing educational choices, and (if the student's conception has changed) how previous conceptions influenced educational decisions.

Understanding a student's conceptions while also judging if the conception is productive or potentially problematic is challenging. The interview must be open-ended enough to allow students to bring up what they believe are the relevant points, but specific enough to test students' understanding. The initial questions will be open ended, with the initial goal of allowing participants to reflect and tell their stories

about how they came to their current understanding of CS. Once a participant articulates a definition of CS, the interviewer will try and ask enough specific questions of the participant to come to an understanding of whether the student's conception of CS is productive or potentially problematic. This highly specific followup is perhaps contrary to usual Grounded Theory interview techniques [10] and more in line with techniques used to get students to reason about scientific experiments [50]. For this project, a cognitive aspect of the interviews is necessary to differentiate between productive conceptions and the many ways that students can maintain problematic conceptions while answering in convincing ways [27]. Appendix A includes an initial interview guide, based on what is known about student conceptions of CS. This guide is expected to change during the course of interviews to reflect a growing understanding of questions that elicit student conceptions well.

Interviews for CS counselors will initially follow a similar method for students: the conceptions counselors hold are interesting to know. Beyond that, the interview will focus on eliciting counselors' theories on students' conceptions: what conceptions counselors think are common in students, how counselors think student conceptions develop, and how counselors think conceptions affect student education decisions.

3.1.4 Analysis to Develop a Grounded Theory

For the more structured aspects of the interviews, analysis should be straightforward. One of the goals of the structured interview processes is to determine if students have a productive or potentially problematic conception of CS; that should be clear from student reasoning about the curriculum and other educational choices. If the interviewer is unable to decide if the students' view of CS made sense, even the most sophisticated Grounded Theory analysis will not be able to answer that question after the fact.

What Grounded Theory can help understand is the common threads that underlie

different student conceptions. Each conception arises from a student’s unique experience. The goal of the analysis is to develop a theory, grounded in each student’s individual experiences, that describes how conceptions develop, change, and influence educational choices.

A grounded theory is based off careful line-by-line analysis of data sources that are methodically abstracted into categories and theories. In this case, the sources include interviews and written materials like CS textbooks and departmental materials. First the researcher develops initial codes that describe what is being expressed in each line of the data [10]. Second, the researcher goes back through the body of research accumulated and selects ‘focused’ codes that explain larger segments of the data. Third, the focused codes are abstracted into categories in a tentative theory that is then checked against other parts of the data to test its explanatory power. There are several techniques to help the researcher attempt to develop the categories in this larger theory including axial coding [62], theoretical coding [10], and situational maps [12]. Tentative theories and ideas are written in memos, which the researcher periodically reviews in order to ensure that nothing significant is missed. The interview/analysis process continues until “saturation”: when additional interviews do not further elaborate the theory.

3.1.4.1 Checks to Ensure Validity

When attempting to understand student conceptions, there is a risk of misinterpretation and bias. This is a common problem in qualitative research; even when participants and researchers act in good faith, it is difficult to understand when backgrounds and assumptions are different. There are a variety of techniques to mitigate this risk [40]. I plan to use two: triangulation from multiple data sources and member checking.

The survey instrument allows me to use triangulation: comparing data from two

similar sources to verify that interpretations of one source are consistent with the other. The survey asks similar questions to the interview process, but helps avoid accidental leading that may occur with expressions and other accidental social cues. Because the survey is written based on the same theories that guide the interview process, some bias can still occur, but places where the survey and interview data diverge will be places to carefully reason about what caused the disparity.

I plan to also use member checking: contacting research participants after the initial interview with the researcher's interpretation of their viewpoint. This gives the participant an opportunity to correct the researcher if they feel their view is misunderstood. Of course it is possible that the researcher notices some pattern that the participant is unaware of; the researcher need not throw out every analysis the participant disagrees with. The researcher must consider participant objections seriously as potential warning signs of misinterpretation or bias. With member checking, one also needs to be aware that participants may be likely to agree with researchers, and therefore not simply take bland agreement at face value ([10], pg. 111).

3.1.5 Analysis of the Preliminary Survey Instrument

The primary goal of the preliminary survey instrument is to generate a reliable way to elicit conceptions that can be used at a larger scale than individual interviews. As the grounded theory develops and common conceptions of CS are identified, the survey will be modified to incorporate questions that work well to elicit key differences in student conceptions. To the extent that the final instrument can delineate between student conceptions and return results that are reliably similar to student interviews, it will be considered successful.

An important question is whether the survey should be administered before or after the interviews with students. When I begin interviewing students, the survey will be administered after the interview process: this provides the least risk of specific

questions on the survey guiding student thinking and corrupting the open-ended interview responses. As the theory is refined and questions that work well with a variety of conceptions have been identified, I will begin administering the survey before the interviews. This lets me test that the survey questions are understandable on their own without the interviewer. In both cases, I will have participants think aloud as they fill out the survey to ensure that the questions are being interpreted as intended.

3.2 Study 2: Assessing Prevalence of CS Conceptions

In Study 2, the survey instrument developed as part of study one is given to students. The goal of this study is to assess how common conceptions elicited in the first study are. Based on what we learn of the way conceptions influence educational decisions in Study 1, this study will suggest how frequently potentially problematic conceptions affect student educational decisions.

3.2.1 Participants

Participants will be students in two different CS classes. The exact classes to be chosen will be decided based on the results of Study 1, but I anticipate mid-sized classes of approximately 30–50 students per class. Participants will take the survey as part of in-class time or in a controlled group environment outside of class to prevent the use of outside materials. In order to encourage students to answer the survey instrument fully, extra credit will be offered for completing the survey.

3.2.2 Analysis

The primary analysis will be attempting to determine student conceptions of CS, based on their survey responses. Student responses will be analyzed by two independent researchers based on a rubric developed from the grounded theory. Cohen's κ will be computed to determine the extent to which the rubric and survey produces

a reliable measure of student conceptions. Depending on what connections between conceptions of CS and educational choices, it may also be possible to attempt to correlate conceptions of CS to grades, future plans, or other external measures of behavior. This larger scale study may also provide evidence for conceptions of CS that were not uncovered in Study 1, or other new qualitative data. If so, a Grounded Theory analysis of the written documents of Study 2 may also be worthwhile.

It is possible that the survey will be unable to clearly delineate conceptions. Although one of the goals of the previous study was to design a successful survey instrument, student conceptions are definitely complex and building reliable survey instruments is difficult. If that happens, hopefully it will be possible to extract a general picture of student conceptions that the two researchers can agree on. The issues of the previous instrument can be examined, with the hopes of resolving them and building a reliable survey as future work.

3.3 Summary

This chapter has outlined the proposed method for two studies to understand student conceptions of CS. In order to know if student conceptions are important from an educational perspective, we need to understand what conceptions exist, if and or how they affect student educational decisions, and how prevalent these conceptions are in students.

Study 1 attempts to discover what student conceptions exist and how they affect student educational decisions. Grounded Theory is selected as the method because of its flexibility to explore unanticipated results and focus on developing a theory true to the understanding of participants. Written materials, interviews with student advisors, and interviews with students will provide the data for the study. Theoretical sampling will be used to determine who to interview as the theory develops. As part of the process, a survey instrument will be developed to elicit conceptions of CS in a

similar way to interviews.

Study 2 uses the survey instrument developed in Study 1 to attempt to understand how prevalent different student conceptions of CS are. Two classes will fill out the survey. Two independent researchers code the conceptions elicited using a rubric based on the theory. If the coding between the researchers are consistent, it should support the idea that the survey is eliciting real conceptions. The prevalence of the various conceptions observed in the classes should give a real idea about the extent to which various conceptions of CS exist in the population.

This chapter outlined the design of two studies to address the research questions put forward in Chapter 1. Detailed examples of the materials for Study 1 can be found in Appendix A. The main body of the proposal concludes in the next chapter, with a proposed timeline for the studies presented here.

CHAPTER IV

PLAN FOR COMPLETION

My overall goal is to defend in December 2011. I plan to do the majority of the interviews and Grounded Theory analysis for Study 1 between January and May 2011. From May 2011 to August 2011, I plan to write the results of that study, do any additional interviews that I discover are necessary, and prepare the final version of my survey instrument and rubric. I hope to distribute the surveys for Study 2 in September 2011. Between September and December I plan to analyze and write up the results of Study 2 as well as my overall conclusions. I consider this an aggressive schedule; depending on the speed of which analysis proceeds the defense may have to move to Spring 2012.

<i>Task</i>	<i>Expected Completion</i>
Proposal Defense	January 2011
Conduct Study 1 Interviews	May 2011
Write Analysis of Study 1	August 2011
Conduct Study 2	September 2011
Write Analysis of Study 2	October 2011
Submit Thesis Draft	November 2011
Thesis Defense	December 2011

APPENDIX A

STUDY 1 MATERIALS

A.1 Initial Interview Guide

Students Script

Rapport-Building Questions - *more ordinary starting places*

1. Tell me about how you chose to major in CS.
2. What has it been like being in the CS program at your school?
3. Tell me about the courses you are taking this semester.
4. What sort of CS courses are you planning on taking in the coming semesters? Tell me about what you expect to learn in these courses.
5. Tell me about your goals after graduation.
6. What do you still need to learn before you can <active goal>?
7. How has <previous course> helped you achieve these goals?
8. How will <future course> help you achieve these goals?

CS Conception Questions - *questions designed to probe understanding of CS*

9. Could you describe some of the most important things you've learned in CS?
10. Tell me about the field of Computer Science.
11. Tell me about how you tell if a particular activity is CS or not.
12. You talked about <course> earlier, why do your instructors consider that material important?
13. You talked about <course> earlier, how does that relate to your overall definition of CS?
14. Do you think of CS as good training for <specific activity>? What parts of CS make it good training for <activity>?
15. You said that ____ is important in Computer Science. What makes you say that?
16. You said that Computer Science is _____. Tell me how you came to that description.
17. Have you ever discussed what CS is with someone else? Tell me about that.

Changing Conception Questions - *questions designed to look at history of CS conception and probe for issues arising from potentially problematic conceptions*

18. Tell me about how your view of CS has developed,
19. Prior to _____, how did you think about CS?
20. Once your view changed from _____ to _____, did you approach things any differently than you had in the past? Can you give me a specific example of something you used to do that you would do differently now?
21. Can you tell me a story that exemplifies how you thought of CS at that time?
22. Who, if anyone, has influenced your view of Computer Science? Tell me how they influenced you.
23. What advice would you give someone similar to you who was starting to major in CS?
24. What other viewpoints about CS have you encountered?
25. How do you think your perspective on CS will change, if at all, as you continue your degree program?

Alignment Questions - *questions designed to check if student feels their view of CS is aligned with the curriculum*

26. If you could change your school's CS degree program, what would you change if anything?
27. Have you ever disagreed with a CS teacher about what was important in a particular course?
28. Imagine I was a new student who wanted to _____ like you. I want you to give me the

real dirt. What should I do? What should I watch out for?

Closing Questions

29. Tell me about how your views on other things have changed since you started studying Computer Science?

30. On the whole, how would you characterize your experiences as a CS major here at <school>?

31. After having these experiences, what advice would you give to someone just starting in CS at <school>?

Is there anything you'd like to ask me?

Teacher/Counselor Script

Rapport-Building Questions

1. What events led up to you becoming a student advisor?
2. What your sorts of concerns do students bring to you most often?
3. When a student comes to you with a concern, what do you think about before you advise them?
4. Do you frequently advise students, who in your opinion, are making poor educational decisions? What, in your opinion, causes students to make these kinds of poor educational decisions?

Conception Questions

5. Tell me how you think about the field of Computer Science.
6. You talked about <course> earlier, how does that relate to your overall definition of CS?
7. You said that ____ is important in Computer Science. What makes you say that?
8. Compare your views with the views of students you talk to most frequently. Do you think that students have a different view of CS than you do?
9. Is that the only view of CS you encounter in students?
10. Can you give me an example of an interaction with a student you had where this was evident?

Changing Conception Questions

11. Tell me about how you think students viewpoints of CS change over time?
12. When students have <conception>, does that affect their educational decisions?
13. Can you think of a specific example of a student with <conception> and how that affected their decisions?
14. When you advise students, do you find it difficult to change their views?
15. What do you think causes students to have <conception>?

Alignment Questions

16. Do you find students find the curriculum does not include things that they feel are important?
17. What kinds of students have the most trouble fitting what they want to do into the way the curriculum is set up?

Closing Questions

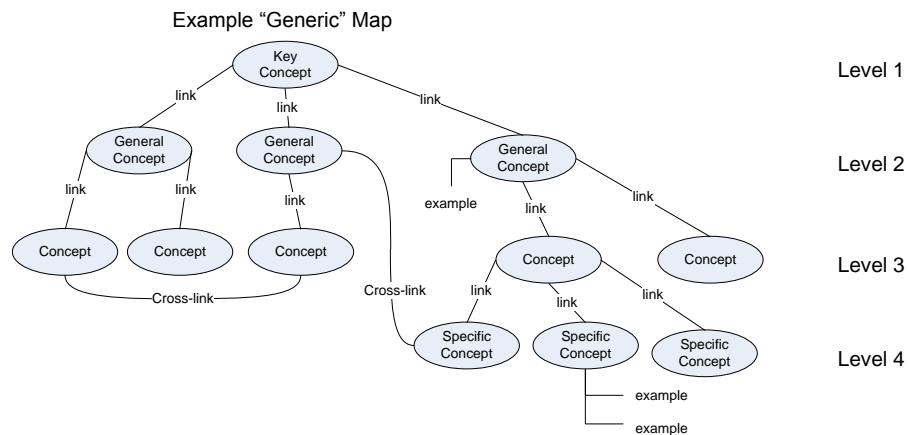
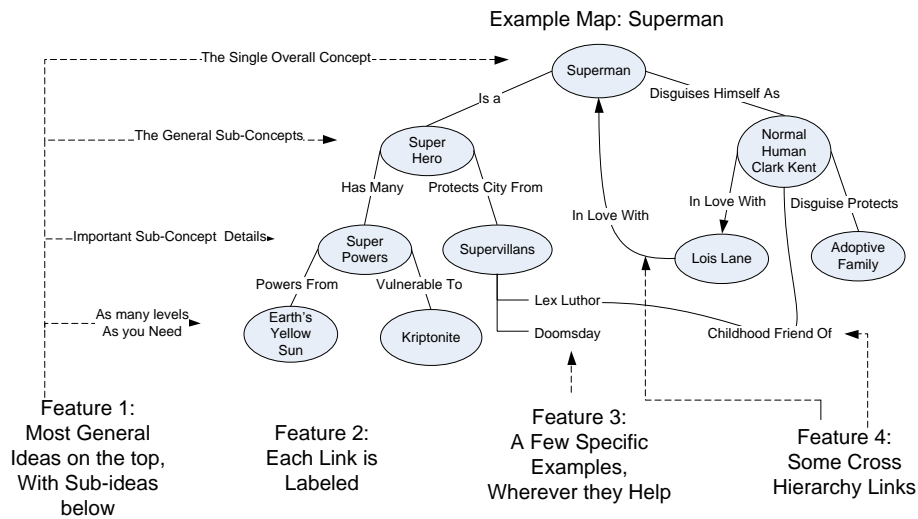
18. Tell me about how your views have changed since you began advising students?
 19. What advice would you like to give to all CS students?
 20. What advice would you give to someone who is beginning to advise students?
- Is there anything you would like to ask me?

A.2 Initial Survey Document

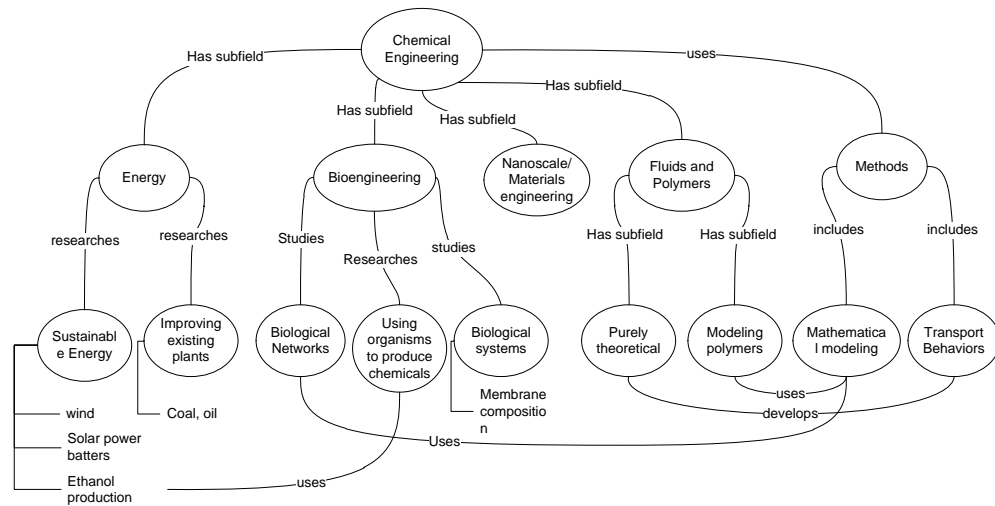
Concept Maps

This survey asks you to build a Concept Map. A concept map is a hierarchical diagram representing a single overall concept, and all the important sub concepts that are part of it. A concept map has 4 key features, summarized in the example below. Before you begin, please look over the examples below and make sure you understand what a concept map is intended to do.

Please look on the next page for a more complex example.



Below you can see an example concept map of the field of Chemical Engineering, as designed by a Chemical Engineering student. Note that there may be many possible correct Concept Maps for a single idea.



How to Build A Concept Map:

1. Decide what concepts you want to include in your map. Try to think about your concepts from a variety of perspectives to generate a lot of different sub-concepts.
3. Arrange these concepts from general to specific
4. Start trying to build your map (I recommend pencil). Note that the first time you try, it'll likely not be right. A good concept map takes several revisions.

Questions to Ask Yourself To See if Your Map Makes Sense:

1. Does the map show hierarchy? Is each subordinate concept more specific and less general than the concept above it?
2. Is the meaning relationship between two concepts indicated by a linking word?
4. Are there a small number of good examples?
3. Does the cross-hierarchy links show meaningful connections between 2 concepts in different parts of the hierarchy?

Name _____

CS Field Survey v1

Part 1

Please draw a *concept map* representing the field of Computer Science. If possible, please include the following items in your concept map: CS Theory, Compilers, Human-Computer Interaction (HCI), and Logic Gates (e.g. AND OR NOT).

Part 2

Please answer these questions.

1. How would you define Computer Science?
2. What would be your ideal job, post-graduation?
3. What areas of CS do you consider best prepare you for the job you selected in question 2 and why?
4. What types of skills/information would you like to see covered in greater detail in your school's CS curriculum?

REFERENCES

- [1] AIKENHEAD, G. S., “Science education: Border crossing into the subculture of science,” *Studies in Science Education*, vol. 27, no. 1, p. 1, 1996.
- [2] ANDERSON, J. R., REDER, L. M., and SIMON, H. A., “Situated learning and education,” *Educational Researcher*, vol. 25, pp. 5–11, May 1996.
- [3] ASTIN, A. W., *What Matters in College: Four Critical Years Revisited*. Jossey-Bass, Jan. 1997.
- [4] BARTON, A. C., *Feminist science education*. Teachers College Press, 1998.
- [5] BIGGERS, M., BRAUER, A., and YILMAZ, T., “Student perceptions of computer science: a retention study comparing graduating seniors with cs leavers,” in *Proceedings of the 39th SIGCSE technical symposium on Computer science education*, (Portland, OR, USA), pp. 402–406, ACM, 2008.
- [6] BRANSFORD, J. D., FRANKS, J. J., VYE, N. J., and SHERWOOD, R. D., “New approaches to instruction: Because wisdom can’t be told,” in *Similarity and analogical reasoning*, p. 470497, 1989.
- [7] BRUCKMAN, A., BIGGERS, M., ERICSON, B., MCKLIN, T., DIMOND, J., DISALVO, B., HEWNER, M., NI, L., and YARDI, S., ““Georgia computes!”: improving the computing education pipeline,” in *Proceedings of the 40th ACM technical symposium on Computer science education*, (Chattanooga, TN, USA), pp. 86–90, ACM, 2009.
- [8] CARTER, L., “Why students with an apparent aptitude for computer science don’t choose to major in computer science,” in *Proceedings of the 37th SIGCSE*

- technical symposium on Computer science education*, (Houston, Texas, USA), pp. 27–31, ACM, 2006.
- [9] CASSEL, L., CLEMENTS, A., DAVIES, G., GUZDIAL, M., McCAULEY, R., McGETTRICK, A., SLOAN, R., SNYDER, L., TYMANN, P., and WEIDE, B., “Computer science curriculum 2008: An interim revision of CS 2001,” *Association for Computing Machinery and IEEE Computer Society*, 2008.
 - [10] CHARMAZ, K., *Constructing Grounded Theory: A Practical Guide through Qualitative Analysis*. Sage Publications Ltd, 1 ed., Jan. 2006.
 - [11] CLANCY, M., “Misconceptions and attitudes that interfere with learning to program,” *Computer science education research*, p. 85100, 2004.
 - [12] CLARKE, A., *Situational Analysis: Grounded Theory After the Postmodern Turn*. Thousand Oaks, Calif: Sage Publications, 2005.
 - [13] CLOUGH, E. E. and WOOD-ROBINSON, C., “Childrens understanding of inheritance,” *Journal of Biological Education*, vol. 19, no. 4, p. 304310, 1985.
 - [14] COMMITTEE FOR THE WORKSHOPS ON COMPUTATIONAL THINKING AND NATIONAL RESEARCH COUNCIL, *Report of a Workshop on the Scope and Nature of Computational Thinking*. National Academies Press, 2010.
 - [15] COMMITTEE ON DEVELOPMENTS IN THE SCIENCE OF LEARNING, *How People Learn: Brain, Mind, Experience, and School: Expanded Edition*. National Academies Press, 2 ed., Sept. 2000.
 - [16] COMMITTEE ON PROSPERING IN THE GLOBAL ECONOMY OF THE 21ST CENTURY, *Rising Above the Gathering Storm: Energizing and Employing America for a Brighter Economic Future*. Washington, D.C.: The National Academies Press, 2007.

- [17] DENNING, P. J., “Is computer science science?,” *Commun. ACM*, vol. 48, no. 4, pp. 27–31, 2005.
- [18] DENNING, P. J., COMER, D. E., GRIES, D., MULDER, M. C., TUCKER, A., TURNER, A. J., and YOUNG, P. R., “Computing as a discipline,” *Computer*, vol. 22, no. 2, pp. 63–70, 1989.
- [19] DENNING, P. J. and McGETTRICK, A., “Recentring computer science,” *Commun. ACM*, vol. 48, no. 11, pp. 15–19, 2005.
- [20] DONALD, J. G., *Learning to Think: Disciplinary Perspectives*. Jossey-Bass, 1 ed., Mar. 2002.
- [21] DWECK, C., *Self-theories: Their Role in Motivation, Personality, and Development*. Psychology Press, 1 ed., Jan. 2000.
- [22] EDWARD, N. S., “Preaching to the converted,” *International Journal of Electrical Engineering Education*, vol. 39, no. 3, p. 230237, 2002.
- [23] ENGLE, C., “Software engineering is not computer science,” *Software Engineering Education*, p. 257262, 1989.
- [24] ERICKSON, G. L., “Children’s conceptions of heat and temperature.,” *Science Education*, vol. 63, pp. 221–30, Apr. 1979.
- [25] FEIL, A. and MESTRE, J., “Change blindness as a means of studying expertise in physics,” *Journal of the Learning Sciences*, vol. 19, no. 4, pp. 480–505, 2010.
- [26] FOUNDATION, W. E. and ACM, “New image for computing: Report on market research.” <http://www.acm.org/membership/NIC.pdf>, 2009.
- [27] GILBERT, J. K., OSBORNE, R. J., and FENSHAM, P. J., “Children’s science and its consequences for teaching,” *Science Education*, vol. 66, no. 4, pp. 623–633, 1982.

- [28] GILBERT, J. K., OSBORNE, R. J., and FENSHAM, P. J., “Children’s science and its consequences for teaching,” *Science Education*, vol. 66, no. 4, pp. 623–633, 1982.
- [29] GREENING, T., “Computer science: through the eyes of potential students,” in *Proceedings of the 3rd Australasian conference on Computer science education*, (The University of Queensland, Australia), pp. 145–154, ACM, 1998.
- [30] HEERSINK, D. and MOSKAL, B. M., “Measuring high school students’ attitudes toward computing,” in *Proceedings of the 41st ACM technical symposium on Computer science education*, (Milwaukee, Wisconsin, USA), pp. 446–450, ACM, 2010.
- [31] HERMAN, G. L., LOUI, M. C., and ZILLES, C., “Creating the digital logic concept inventory,” in *Proceedings of the 41st ACM technical symposium on Computer science education*, (Milwaukee, Wisconsin, USA), pp. 102–106, ACM, 2010.
- [32] HEWNER, M. and GUZDIAL, M., “Attitudes about computing in postsecondary graduates,” in *Proceeding of the fourth international workshop on Computing education research*, (Sydney, Australia), pp. 71–78, ACM, 2008.
- [33] HEWNER, M. and KNOBELSDORF, M., “Understanding computing stereotypes with Self-Categorization theory,” in *Proceedings of Koli Calling International Conference on Computer Science Education*, (Koli National Park, Finland), 2008.
- [34] JAMES, R., “How school-leavers chose a preferred university course and possible effects on the quality of the school-university transition,” *Journal of Institutional Research*, vol. 9, no. 1, p. 7888, 2000.
- [35] KACZMARCZYK, L. C., PETRICK, E. R., EAST, J. P., and HERMAN, G. L., “Identifying student misconceptions of programming,” in *Proceedings of the 41st*

- ACM technical symposium on Computer science education*, (Milwaukee, Wisconsin, USA), pp. 107–111, ACM, 2010.
- [36] KUHN, D., “Children and adults as intuitive scientists,” *Psychological review*, vol. 96, no. 4, p. 674, 1989.
 - [37] KUHN, D., “Science as argument: Implications for teaching and learning scientific thinking,” *Science Education*, vol. 77, no. 3, p. 31937, 1993.
 - [38] LAVE, J. and WENGER, E., *Situated Learning: Legitimate Peripheral Participation*. Learning in doing, Cambridge, UK: Cambridge University Press, 1991.
 - [39] LEWIS, T. L. and SMITH, W. J., “The computer science debate: it’s a matter of perspective,” *SIGCSE Bull.*, vol. 37, no. 2, pp. 80–84, 2005.
 - [40] LINCOLN, Y. S. and GUBA, E. G., *Naturalistic inquiry*. SAGE, 1985.
 - [41] LINN, M. C. and MUILENBURG, L., “Creating lifelong science learners: What models form a firm foundation?,” *Educational Researcher*, vol. 25, pp. 18 –24, June 1996.
 - [42] MALONEY, J. H., PEPPLER, K., KAFI, Y., RESNICK, M., and RUSK, N., “Programming by choice: urban youth learning programming with scratch,” *SIGCSE Bull.*, vol. 40, no. 1, pp. 367–371, 2008.
 - [43] MARGOLIS, J., *Stuck in the Shallow End: Education, Race, and Computing*. The MIT Press, Sept. 2008.
 - [44] MARGOLIS, J. and FISHER, A., *Unlocking the Clubhouse: Women in Computing*. The MIT Press, Apr. 2003.
 - [45] MARTIN, C. D., “The case for integrating ethical and social impact into the computer science curriculum,” in *The supplemental proceedings of the conference*

- on Integrating technology into computer science education: working group reports and supplemental proceedings*, (Uppsala, Sweden), pp. 114–120, ACM, 1997.
- [46] MCGUFFEE, J. W., “Defining computer science,” *SIGCSE Bull.*, vol. 32, no. 2, pp. 74–76, 2000.
 - [47] MYERS, D., “Self-Serving bias,” in *Social Psychology*, pp. 61–69, McGraw-Hill Humanities/Social Sciences/Languages, 9 ed., Dec. 2006.
 - [48] NESPOR, J., *Knowledge in motion: Space, time, and curriculum in undergraduate physics and management*. Routledge, 1994.
 - [49] NOVAK, J. D. and GOWIN, D. B., *Learning How to Learn*. Cambridge University Press, 1 ed., Sept. 1984.
 - [50] OSBORNE, R. J. and GILBERT, J. K., “A technique for exploring students’ views of the world,” *Physics Education*, vol. 15, no. 6, pp. 376–379, 1980.
 - [51] PAPERT, S., *Mindstorms: children, computers & powerful ideas*. Prentice Hall / Harvester Wheatsheaf, 1982.
 - [52] RASMUSSEN, B. and HAPNES, T., “Excluding women from the technologies of the future? a case study of the culture of computer science,” *Futures*, vol. 23, no. 10, pp. 1107–1119, 1991.
 - [53] ROSENBLOOM, P. S., “A new framework for computer science and engineering,” *Computer*, vol. 37, no. 11, p. 2328, 2004.
 - [54] RUSHKOFF, D., *Program or be Programmed*. OR Books, 2010.
 - [55] SCHULTE, C. and KNOBELSDORF, M., “Attitudes towards computer science-computing experiences as a starting point and barrier to computer science,” in *Proceedings of the third international workshop on Computing education research*, (Atlanta, Georgia, USA), pp. 27–38, ACM, 2007.

- [56] SHACKELFORD, R., MCGETTRICK, A., SLOAN, R., TOPI, H., DAVIES, G., KAMALI, R., CROSS, J., IMPAGLIAZZO, J., LEBLANC, R., and LUNT, B., “Computing curricula 2005: The overview report,” *SIGCSE Bull.*, vol. 38, no. 1, pp. 456–457, 2006.
- [57] SHAMOS, M. H., *The myth of scientific literacy*. Rutgers University Press, 1995.
- [58] SHERMAN, R. R. and WEBB, R. B., *Qualitative research in education: Focus and methods*. Routledge, 1988.
- [59] SMITH, J. P., DISESSA, A. A., and ROSCHELLE, J., “Misconceptions reconceived: A constructivist analysis of knowledge in transition,” *Journal of the Learning Sciences*, vol. 3, no. 2, pp. 115 – 163, 1994.
- [60] SONGER, N. B. and LINN, M. C., “How do students’ views of science influence knowledge integration?,” *Journal of Research in Science Teaching*, vol. 28, no. 9, pp. 761–784, 1991.
- [61] STEVENS, R., OCONNOR, K., GARRISON, L., JOCUNS, A., and AMOS, D. M., “Becoming an engineer: Toward a three dimensional view of engineering learning,” *Journal of Engineering Education*, vol. 97, no. 3, p. 355368, 2008.
- [62] STRAUSS, A. C. and CORBIN, J., *Basics of Qualitative Research: Second Edition: Techniques and Procedures for Developing Grounded Theory*. Sage Publications, Inc, 2nd ed., Sept. 1998.
- [63] SUDOL, L. A. and JASPAN, C., “Analyzing the strength of undergraduate misconceptions about software engineering,” in *Proceedings of the Sixth international workshop on Computing education research*, (Aarhus, Denmark), pp. 31–40, ACM, 2010.

- [64] TEW, A. E., FOWLER, C., and GUZDIAL, M., “Tracking an innovation in introductory CS education from a research university to a two-year college,” *SIGCSE Bull.*, vol. 37, no. 1, pp. 416–420, 2005.
- [65] TEW, A. E. and GUZDIAL, M., “Developing a validated assessment of fundamental CS1 concepts,” in *Proceedings of the 41st ACM technical symposium on Computer science education*, (Milwaukee, Wisconsin, USA), pp. 97–101, ACM, 2010.
- [66] WANDERSEE, J. H., MINTZES, J. J., and NOVAK, J. D., “Research on alternative conceptions in science,” *Handbook of research on science teaching and learning*, p. 177210, 1994.
- [67] WING, J. M., “Computational thinking,” *Communications of the ACM*, vol. 49, no. 3, pp. 33–35, 2006.
- [68] YARDI, S. and BRUCKMAN, A., “What is computing?: bridging the gap between teenagers’ perceptions and graduate students’ experiences,” in *Proceedings of the third international workshop on Computing education research*, (Atlanta, Georgia, USA), pp. 39–50, ACM, 2007.