I have two goals for this paper. The first is to “go meta”, to question the very research questions that we are to generate, by unpacking their presuppositions. These suppositions are embedded in the language that we use to describe our studies, and carry through the entire design, the implementation, the interpretation, and the reporting of our results. The problem with unquestioned presuppositions, is that they can prevent our community from looking outside of standard ways of carrying out our collective work, continuing business as usual, research as we have been doing it. And one of my concerns with this is that we are not in sufficient discourse with other educational researchers, other disciplines. The problems of being relatively insular are twofold. First, we are not bringing in, harvesting, appropriating what others are doing, and therefore advancing the field with those innovations that are happening elsewhere. And second, we are not giving out, and thereby establishing that CS is legitimate domain where issues of teaching and learning share some but not all of the characteristics of teaching and learning in other domains, and also where some of the issues in CS are particular to the domain.

My plan, then, reflects these two goals. I start by taking some of the questions that were referenced in the solicitation for participation in this summit, and unpack the presuppositions embedded there. I then ask a new set of questions, based on a different set of presuppositions. And, as with the original set, I make such presuppositions explicit. The point is not to say “my presuppositions are better than yours”, that A is better than B, but to say that at our point in development as an interdiscipline, it is premature to take our practices, and what they embed and entail, as given. There are credible other kinds of presuppositions (many of which are beyond what I can express, because I am biased by my presuppositions), and these will lead to different kinds of questions and research studies.

I begin with some of the questions that Mark Guzdial posted on his blog (Guzdial, 2012). I choose Mark’s questions for a few reasons. One is that I have great respect for Mark, and believe that he wants to stimulate dialog. Another is that Mark and I have already had discussions on some of the ideas discussed in this white paper and in this summit, and as result Mark himself bears some responsibility (though he might not have known it, or might disavow it on seeing my position) in the production of these ideas. And a third, I think that they are a good set of ideas, worth taking seriously. I do not intend to criticize Mark’s beliefs, but rather to explore some of their unspoken assumptions. I come to praise Mark, not to bury him.

*Question: What are the cognitive processes of learning to program?*

In talking about “cognitive processes”, Mark references a language about the mind that can be termed cognitivism. Cognitivism has been the dominant paradigm in Psychology since the latter half of the 20th century. Many date the start of the “cognitive revolution” in Psychology to the late 1950’s (Miller, 2003), when researchers began to study the contents of mind—goals, intentions, plans, mental representations. Under the behaviorist view that predominated Psychology throughout much of the early part of the 20th century, such mentalistic entities were considered unscientific to study because of their subjective nature and the impossibility of directly observing them (Bruner, 1990; Miller, 2003). The cognitivists posit that the mind is like a logic machine, carrying out computation on the contents of memory. As one of the founders of the cognitive sciences comments, these researchers “dream of a unified science that would discover the
Knowledge as represented in the mind is viewed as context-free, allowing for formal rules of inference to be used to generate new understandings inferred from the symbol structures. And it is the abstract and context-free nature of knowledge that allows researchers in this tradition to treat different structures for encoding knowledge (such as the nodes and links of a semantic network, the slots and slot fillers of a frame, or the sentences in logic) as equivalent. Prior to action in the world, a cognitive agent (whether human or artificial) is hypothesized to think through the consequences of action purely through syntactic operations on the mental representation. The thinking agent thus generates counterfactual statements about possible future states of affairs (“if I were to do X then Y would occur”) so as to construct plans of action to bring about desired world states. Viewing minds as symbolic machines that carry out inference procedures on symbolic representations leads to a particular view of learning: learning is the acquisition, change, and application of symbol structures that denote knowledge about the world, such as scripts, plans, and schemas.

In using this language, then, Mark views the individual mind as the unit of analysis, and the mental, what is inside “skin and skull,” as the focus. “Learning to program”, then, primarily concerns the acquisition of programming plans or schemas, internal representations of the computational domain that organize it in a general and context-free fashion for the individual. Such representations are then retrieved from memory when individuals face new problems (e.g. having to write a program to solve a particular problem), and applied in a relatively straightforward fashion to a concrete problem at hand.

This first question can be read in light of what Mark himself defines as his fundamental conception of what “computing education” is, which, by reflection, can be read in light of this question. He states: “Computing education research is the study of how people come to understand computing.” Computing education is thus fundamentally concerned with people's understanding of computing. The Oxford English Dictionary provides 35 senses of understand (The Oxford English Dictionary, n.d.). I emphasize several of these senses here due to their concern with “the cognitive processes of” or similar. These are:

1a) To comprehend; to apprehend the meaning or import of; to grasp the idea of.
2a) To comprehend by knowing the meaning of the words employed; to be acquainted with (a language) to this extent.
3a) To comprehend as a fact; to grasp clearly, to realize.
4a) To grasp as a fixed or established fact or principle; to regard as settled or implied without specific mention.
4b) To have knowledge of, to know or learn, by information received.
5) To take, interpret, or view in a certain way.
11) To have knowledge or information, to learn, of something.

As can be seen, these are not all of the readings that one can make of the word, and I return to some later that I have not included here, senses more closely associated with different suppositions. One might question the inclusion of 4a, concerning “established facts.” But this sense of “understand” is clearly present in Mark's 2nd research question: “Where do problems/difficulties/misconceptions in learning programming come from?” To have misconceptions is to presuppose fixed or established facts or principles; it is only with respect to these that we can even talk about misconceptions.
I wish to underscore (again), that I am not critiquing Mark, nor am I challenging this conception of computing education and the research questions that this conception gives rise to. My first goal is to make the cognitivist presuppositions that are immanent in some of his questions visible. At the same time, I also want to point out that Mark has other questions that have different presuppositions that I do not pursue here (but that nonetheless might be revealed by others), concerning motivation, particular means for teaching computer programming, teacher education, and others. That is, despite the presentation above, I do not want to suggest that Mark only believes the presuppositions stated above. In addition, I could just as well have chosen the questions posed in the solicitation for this workshop, and looked at some of their presuppositions (e.g. “How do students learn computer science concepts, such as programming?” “How can we measure student learning in computer science classes?”), some of which share many of the suppositions that I have explored here.

What, then, are different presuppositions from which we might start, and what are some of the questions that these give rise to? I first provide a few questions, and then describe the different presuppositions that they reflect.

- How do two software developers working side by side to achieve a joint goal use the resources in their environment and that they themselves generate in order to achieve mutual intelligibility?
- How do groups of software developers use externalized representations of computational systems in the design of these very systems?
- How do experienced software developers within a particular setting make their practices public so that they can be appropriated and reshaped by newcomers to the practice?

Note, first, that none of the questions mentions “students” specifically. This embeds a presupposition that learning within formal settings is not different, in some fundamental ways, from learning in other settings, such as the workplace, the home, the church, the street corner. This comes from a view of learning best articulated by Barbara Rogoff: “Humans develop through their changing participation in the sociocultural activities of their communities, which also change” (2003, p. 368). Culture is not something that is static, but is produced by the actions of those who share the culture. And the culture changes even as it produced and reproduced. What this means is that schools and universities are not somehow acultural; they do not stand outside of the same kinds of developmental processes that occur as individuals participate in social groups. Schools and universities develop their own cultural forms of activity, of practice, and have particular resources at hand. All of which means (among other things) that students may learn “school programming” or “school computer science”, and yet abandon these cultural forms of participation when they enter different settings with different resources and cultural practices, much as the dairy workers (Scribner, 1999), fish hatchery workers (Roth, 2005), and weight watchers (Lave, 1988) all abandoned “school mathematics” for contextually-specific forms of mathematics that exploited the existing (and changing) resources and practices within their respective settings. In addition, not only do schools and universities become legitimate sites for research (and the students, teachers, and administrators who staff them as legitimate objects of study), but so do any other sites in which the activities of interest occur. These include such places mentioned above: the workplace, the home, the church, the street corner.

Another presupposition in the above research questions emphasizes the crucial role of setting, insofar as settings make available to those within them (and who, at the same time, co-produce these settings) various resources that can be used to mediate, structure, represent, afford, and constrain the activity that occurs within the setting. These can include such things as
whiteboards, computers, meeting rooms, and display projectors, many of which are the products of intentional design by cultural predecessors for the very purpose of facilitating the activity that normally occurs in the setting. But many of the resources within settings are those produced by the people who staff these settings, contingently as they respond to the unfolding action around them. These include such things as gestures and speech, as well as notes and diagrams on whiteboards, and screenfuls of code on computer displays.

And perhaps the most important presupposition in this latter set of research questions is a view of learning as an indecomposable social act. That is, learning (and hence “education”) is not viewed as something that exists “in” individuals, in their minds, specifically. Rather, learning and education are fundamentally wrapped up in activity, in social activity in particular. This is not a denial of knowledge and mind, but rather that they come into being through activity. Knowledge not as something static and possessed, but something active, enacted, and distributed across people and resources.

Note that the presuppositions here, then, do not conceive of learning (solely) as “understanding” - or if so, then only in the sense of the OED’s definition 1b) concerning practice: “To be thoroughly acquainted or familiar with (an art, profession, etc.); to be able to practise or deal with properly”. Or its social sense in 2c) “to understand each other, to be in agreement or collusion; to be confederates.” Or its reflexive use in definition 1d) “To know one’s place, or how to conduct oneself properly”.

Perhaps the largest implication of these presuppositions for education research is a change in the unit of analysis, from that of the atomic individuals, to social groups within particular settings. One can see this in comparing the two sets of questions, the one concerned with individual conceptions and misconceptions, the latter with social activity within particular settings. In addition, these presuppositions mean that settings always exist, and learning is always social. The think-aloud of the cognitivist that happens in the lab, by virtue of its acontextuality, is often seen as generalizable to all settings since the research is not biased by the effects of a particular setting. But this ignores the fact that a lab itself is a particular setting, and the presence or absence of particular people and resources within this setting will have a considerable effect on the activity carried out within it. In addition, the social relationship of the individual being studied and the researcher is never neutral, because the way in which the research subject construes the setting and the social expectations of appropriate behavior within that setting will likewise affect the activity carried out (Ross & Nisbett, 1991).

Returning now to the original goals that I mentioned at the start of this white paper. It is important that we ask research questions, and that we carry out some of the studies that will inform these; how else will we develop a science that successively builds on itself? And yet the presuppositions that we have concerning such things as the nature of mind and activity will constrain and shape the questions that we ask, just as the questions that we ask embed the presuppositions and constrain the studies that we undertake to answer these questions. Changing presuppositions at the same time changes the questions that are asked, and therefore the methods that will be used to study these questions. I am not stating that one set of presuppositions is “better” in any absolute sense than any other—though I confess an affinity for the latter set of presuppositions over the former. My point is that having a clearer, more explicit understanding of our presuppositions (surely a mentalistic concept if there ever was one!) in asking our research questions and carrying out our research is important at this stage in our development. It is important chiefly for two reasons. The first is that without such reflective awareness, we risk routinizing our work, overly constraining it to a set of well-worn paths, to the
kind of science that Thomas Kuhn characterizes as “mopping up” (Kuhn, 1962). But mopping up does not bring with it paradigm shifts, the asking of big questions, the challenging of the very reasons why we enact teaching and learning the way that we currently do. And the second is that CS Education does not exist out of historical time or isolated from the discourse in other disciplines, other fields. It is from such discourses that we can challenge existing presuppositions and appropriate new ones, and in so doing actively participate in this discourse that extends far beyond our boundaries, artificial as they are. And it is the reflective stance of some of these other disciplinary practices that we can appropriate as well. In so doing, we can hopefully enrich and improve CS Ed research, and make it useful for practice.


