Cog468 "Cognitive Science Capstone Seminar"
First Notes Assignment

PASS 3

PRIMARY SOURCES

1. **Practices of Distributed Intelligence and Designs for Education**

    1. While providing few answers, I hope to provoke new questions and inquiries, for distributed intelligence is not a theory of mind, or culture, or design, or symbol systems and their impact on human thought so much as it is a heuristic framework for raising and addressing theoretical and empirical questions about these and other topics. (p. 48)

    2. Activity is achieved in means-end adaptations. These adaptations may be more or less successful. The focus in thinking about distributed intelligence is not on intelligence as an abstract property or quantity residing in minds, organizations, or objects. In its primary sense here, intelligence is manifest in activity that connects means and ends through achievements. (p. 50)

    3. To sum up, knowledge is often carried in artifacts as diverse as physical tools and notational systems such as algebraic equations. This knowledge may come to be exploited in activity by a new learner through a variety of genetic paths: through observations of use by other humans and attempts to imitate it, through playful discovery of its affordances in solitary activity, and through guided participation in its use by more knowledgeable others. And the affordances of such artifacts may be more or less difficult to convey to novice users of these artifacts in the activities to which they contribute distributed intelligence. (p. 54)

2. **Person-Plus: A Distributed View of Thinking and Learning**

    1. Second, the best use of these physical support systems is an art. It is not so commonly found. And conventional instruction does little to acquaint students with this art, mistakenly expecting the fingertip effect to do the job. (p. 96)

    2. Certainly there is justification for some concern with the person-solo. But so much seems quite misguided, for at least two reasons: (1) If part of the mission of schools is to prepare students for out-of-school performance, this perseveration on the person-nearly-solo is not "lifelike"; (2) most students have much to learn about the art of distributed cognition, and schools should help. (p. 95)

    3. The cause of overemphasis on the person-solo may be this. There is a widespread belief in what I have previously called the "fingertip effect": Simply make a support system available and people will more or less automatically take advantage of the opportunities that it affords (Perkins, 1985). Were the fingertip effect a reality, there would be little need for education to worry about students learning to make the best use of supporting environments -- ones as simple as pencil and paper or as complex as word processor, outliner, or hypertext environment. (p. 95)

3. **No Distribution Without Individuals' Cognition: A Dynamic Interactional View**

    1. Specifically, the general hypothesis would be that the "components" interact with one another in a spiral-like fashion whereby individuals' inputs, through their collaborative activities, affect the nature of the joint, distributed system, which in turn affects their cognitions such that their subsequent participation is altered, resulting in subsequent altered joint performances and products. (p. 122)

    2. For both pragmatic and normative reasons, one ought to see in situations entailing distributed cognitions not only ends in themselves but opportunities for the development of cognitive residues that might serve students when on their own. (p. 131)

    3. In order to enter into an intellectually useful partnership with peers or with computer tools, in order for one's activity to undergo transformations, transformations that come about through the distribution of cognitions, one must have certain competencies and proclivities that themselves are developed through the practice provided by the partnership. Competency, as Olson (1986) points out, is "a product of the relationship between the structures of the mind and the properties of 'technologies of the intellect'" (p. 351). (p. 126)

4. **Teaching the Nintendo Generation to Program**

    1. But it's through our use of Squeak and watching students rise to the challenge of multimedia programming that we came to the realization that multimedia-first is a viable way to *introduce* computing. (p. 19)

    2. Guzdial & Soloway (2002) say: Today's desktop computers were invented to be multimedia composition and exploration devices. The Xerox PARC Learning Research Group believed in a vision of the computer as a *Dynabook*: A tool for learning, through creation and exploration of a wide range of media. Pursuing that vision is what led them to invent the desktop user interface as part of their programming language, Smalltalk. Alan Kay and Adele Goldberg spelled out their vision of the Dynabook in a 1977 paper, *Personal Dynamic Media*, that talked about students building music, animation, and drawing systems -- learning to program through the creation of media and learning to program in order to create media. Creating media sounds like what the Nintendo Generation is looking for. (p. 18)

    3. One reason for not introducing programming via multimedia construction is the lack of a good multimedia programming platform. ... This lack of support for multimedia might be due to the perception that multimedia programming is an advanced topic, something that CS1 students might one day aspire to. No one does multimedia *first* ... But, if the platform does support multimedia - as does Alan Kay et al's *Squeak* - multimedia programming can fit in well within the scope of a CS1 course. (pp. 18, 19)

5. **Wise-Up: The Challenge of Lifelong Learning**

    1. I have used these vignettes, each of them based on research which will be discussed later, to introduce the broad scope and some of the main themes of this book - in particular what we might call the three Rs of learning power: resilience, resourcefulness and reflectiveness. (p. 6)

    2. There are four main compartments to the learning toolkit. First, there is direct *immersion* in experience and the practical tools of exploration, investigation and

experimentation that go with it. Some of this kind of learning focuses on the physical world, but much of it is social, involving interaction and imitation, the principal media through which people pass on their practical skills to each other. Then there is *imagination* and the skills of fantasy, visualization and story-telling that enable you to create and explore hypothetical worlds. Next come all the *intellectual skills* of language and reasoning, through which experience can be segmented, analyzed and communicated. And finally there is *intuition* a general name for the family of softer, more receptive processes whereby creative ideas are germinated and developed. (p. 11)

3. Learning, ..., is what you do when you don't know what to do. Learning to learn, or the development of learning power, is getting better at knowing when, how and what to do when you don't know what to do. (p. 11)

6. **Learning and Teaching Programming: A Review and Discussion**

1. Effective novices are those that learn, without excessive effort or assistance, to program. Ineffective novices are those that do not learn, or do so only after inordinate effort and personal attention. It may be productive, in an introductory programming course, to explicitly focus on trying to create and foster effective novices. In other words, rather than focusing exclusively on the difficult and end product of programming knowledge, it may be useful to focus at least in part on the enabling step of functioning as an effective novice.

2. Green (1990, p 117) suggests that programming is best regarded not as "transcription from an internally held representation," or in the context of "the pseudo-psychological theory of 'structured programming'," but as an exploratory process where programs are created "opportunistically and incrementally." A similar conclusion is reached by Visser (1990) and by Davies:

   > ...emerging models of programming behavior suggest an incremental problem-solving process where strategy is determined by localized problem-solving episodes and frequent problem re-evaluation. (Davies, 1993, p. 265)

   An emphasis on opportunistic exploration seems particularly appropriate when considering novice programming.

3. Ideally course design and teaching would take place in the context of familiarity with the key issues that have been identified in the literature. The most basic factor, especially given the observations regarding the limited progress made by novices in introductory courses, is that a CS1 course should be realistic in its expectations and systematic in its development: "Good pedagogy requires the instructor to keep initial facts, models and rules simple, and only expand and refine them as the student gains experience" (Winslow, 1996, p. 21). duBoulay et al. (1998) make a case for the use of simple, specially designed teaching languages. In many cases the role of the course in the broader teaching curriculum may rule this out as an option, and complex "real" languages are typically used. (p. 157)

7. **Will There Ever Be Consensus on CS1?**

1. The choice of programming language, the approach by which students are taught and the software tools made available to students have been controversial issues in many ways. While there once was a consensus of some sort within the computer science education community, it is much more difficult to find command ground amount those

of us who teach introductory programming courses. (p. 1 of my pdf)

2. While it is generally agreed upon that input/output, primitive data types, encapsulation and methods, conditional and iterative statements and documentation should be taught in a CS1 class, the other topics are frequently omitted. Two of the most important skills in the above list are programming/problem-solving skills and adaptive programming skills, which every student must learn. Frequently, a CS1 course in Java contains so much material that these two most important skills can be pushed aside. It is easy to assume that students with an understanding of mathematics that is at least on the lvel of algebra can adapt these problem solving skills to computing; however there is a body of evidence that strongly suggests that this is not the case. Biggs' structure of the observed learning outcome (SOLO) taxonomy discusses such issues [31,32]. It illustrates how students will not learn a topic unless the instructor aligns course objectives to learning outcomes and demonstrates to students how to perform the necessary tasks. (p. 4 of my pdf)

3. McIver recognized the importance of understanding how the novice perceives text written in a particular programming language if we are to evaluate the suitability of any programming language in teaching novices how to program. Green [30] identified thirteen cognitive dimensions of notation. These are, in reality, properties that notations or language may possess that will either make it easier or harder for novices to learn them. McIver found that six of these are of particular importance in evaluating programming languages for use by novice programmers:
    1. "Closeness of mapping" addresses how well the notation represents the domain for which it is attended, e.g., if we are trying to describe arithmetic, how closely does our notation resemble arithmetic?
    2. To be "consistent," similar semantics should be expressed in similar syntax. Therefore, an if..elseif..else construction would be considered more consistent than a switch statement.
    3. "Diffuseness" refers to the verbosity of the language. COBOL would be an example of a diffuse notation.
    4. "Error-prone" constructions are those that are more likely to lead to errors, or perhaps even encourage them. The use of separate pairs of brackets for different dimensions of an array might be considered error-prone.
    5. "Hard mental operations" would require the programmer to prefer potentially difficult tasks in writing a program, e.g., entering all numeric constants in an unusual number base.
    6. "Role expressiveness" refers to the ability of a reader to infer the usage of a feature just from its structure.

    McIver examined several languages, including Java, which failed to meet the optimal case for cognitive dimensions.

8. **A Brief Tour of the Learning Sciences Via a Cognitive Tool for Investigating Melodic Phenomena**

    1. The emphasis in this article is on MxM as a cognitive tool (Jonassen & Carr, 2000), a technology that affords learners enhanced opportunities to engage in processes of knowledge construction and reflective thinking. (p. 182)

    2. From a practical point of view, use of this cognitive tool will enrich the learner's appreciation of music, as a side effect of engaging in generative processes of analysis and composition. According to Whitehead (2002), "Art is the imposing of a pattern on experience, and our aesthetic enjoyment is recognition of the pattern." The computational modeling framework and the corresponding conceptual modeling

methodologies discussed in this article were designed to enhance the modeler's sensitivity to melodic pattern, and hence their aesthetic enjoyment of music. From a theoretical point of view, this tool provides opportunities to gain a better understanding of the cognitive processes associated with music and to sharpen one's insights into generic processes of learning. (p. 183)

3. Specifically, by explicitly relating elements of MxM and examples of its use to a selection of ideas from the learning sciences, I intend to contribute fragments of concrete understanding to the on-going search for meaning in these ideas. (p. 183)

SECONDARY SOURCES

1. **Invent to Learn: Making, Tinkering, and Engineering in the Classroom**

    1. When we talk about a "project," what we mean is work that is substantial, shareable, and personally meaningful. Some projects may take a class period or two to complete, while others may require an entire term. (p. 57)

    2. When a teacher creates a well-designed prompt that capitalizes on student curiosity, kids can embark on complex, long-term learning adventures. (p. 57)

    3. In *Epistemological Pluralism and the Revaluation of the Concrete*, Sherry Turkle and Seymour Papert argue that equal access to mathematics and science (including computer science) for women is not just a matter of historical gender inequity, but a basic imbalance in valuing only "abstract, formal, and logical" ways to think about science.

        > The concerns that fuel the discussion of women and computers are best served by talking about more than women and more than computers. Women's access to science and engineering has historically been blocked by prejudice and discrimination. Here we address sources of exclusion determined not by rules that keep women out, but by ways of thinking that make them reluctant to join in. Our central thesis is that equal access to even the most basic elements of computation requires an epistemological pluralism, accepting the validity of multiple ways of knowing and thinking. (Turkle & Papert, 1991)

        (p. 37)

2. **Distributed Cognition**

    1. The form of distributed cognition advanced by Hutchins (DCog) has adapted the framework of individual cognition to explain how cognitive resources are organized \textsl{within a context}, drawing on actors and other features in the environment to perform problem solving. Hutchins calls this "socially distributed cognition" (1995a). Socially distributed cognition describes group activity in the way that individual cognition has traditionally been described -- computation realized through the creation, transformation, and propagation of representational states (Hutchins, 1995a; Simon, 1981). Central to this is the idea of work being distributed over a range of media and over a number of people. It is concerned with representational states and the informational flows around the media carrying these representations. The DCog

framework allows researchers to consider all of the factors relevant to the task, bringing together the people, the problem, and the tools used into a single unit of analysis. This makes it a suitable candidate for developing an understanding of how representations act as intermediaries in the dynamically evolving and collaborative processes of work activities. (p. 196)

2. Arguably, one of the most important abilities that humans have is the ability to create and use tools, or artifacts (manmade or modified objects; Cole, 1990). One particular subset of these artifacts are those that aid cognition -- known as \textsl{cognitive artifacts} (see Norman, 1991). Cognitive artifacts include external representations of "knowledge in the world" (Norman, 1988) as memory aids, such as checklists, books, diaries, and calendars. However, they are also used to augment information processing itself, not just in increasing memory capacity. A fundamental feature of cognitive artifacts is that they do not simply augment or "amplify," existing human capabilities. Rather, they transform the task into a different one (Cole, 1990; Norman, 1993), allowing resources to be reallocated into a configuration that better suits the cognitive capabilities of the problem solver. (p. 201)

3. As a consequence of the overly reductionist stance of traditional cognitive science and HCI research that ignores cognitive artifacts and social interaction, a number of theoreticians have moved toward what Norman (1993) calls the "systems perspective." Such systems-based perspectives attempt to describe all of the features active in a work or activity system: people, artifacts, and, importantly, the means of organizing these into a productive unit. For these researchers, it is the system, rather than the cognitive properties of an individual or the design of an artifact, that determines overall performance at a task. (p. 201)

3. **Twilight of the Lecture**

    1. But ultimately, learning is a *social* experience. Harvard is Harvard not because of the buildings, not because of the professors, but because of the *students* interacting with one another. (p. 27)

    2. Interactive learning triples students' gains in knowledge as measured by the kinds of conceptual tests that had once deflated Mazur's spirits, and by many other assessments as well. It has other salutary effects, like erasing the gender gap between male and female undergraduates. "If you look at incoming scores for our male and female physics students at Harvard, there's a gap," Mazure explains. "If you teach a traditional course, the gap just translates up: men gain, women gain, but the gap remains the same. If you teach interactively, *both* gain more, but the women gain disproportionately more and close the gap." Though there isn't yet definitive research on what causes this, Mazure speculates that the verbal and collaborative/collegial nature of peer interactions may enhance the learning environment for women students. (p. 24)

    3. "We want to educate leaders, the innovators of society," Mazur says. "Let's turn our students into *real* problem solvers. In a real world problem, you know where you want to get, but you don't know how to get there. For example: how can I bake a cake with no flour? The goal is known, but the prescription to get there isn't. Most tests and exams at Harvard are not like that; they are questions where you need to determine what the *answer* is. In physics it might be, What was the velocity of the car before it hit the tree? There, you know exactly what you need to do: you *have* a prescription to calculate velocity, but you don't know the velocity. It's the opposite of a real-life problem, because you know the prescription, but you don't know the answer." (pp. 26, 27)

4. **Learning and Cognition: The Design of the Mind**

    1. At the pinnacle of Bloom's Taxonomy is the cognitive process of *evaluation*. Why did Bloom choose evaluation as the highest of all cognitive processes? If ideas vary in quality, then to evaluate ideas is to confront that reality directly. The person who evaluates is the one who continually tests ideas for their soundness, workability, defensibility, power, insight, subtlety, and novelty -- for their potential to construct or subvert beliefs, to reconcile or to inflame conflicts, to build knowledge edifices or to kick down the foundations of what we think we know. As one manifestation of metacognition, evaluation is crucial to the pursuit of good ideas and sound plans. Adept thinkers know that knowledge-building is a human enterprise, so declared truth is never definitive, never beyond doubt, never perfect. We evaluate so that we can test and, if possible, improve what we know. This is the essence of complex cognition: forever evaluating and evolving ideas toward something better. (pp. 142-143)

    2. If there is something that binds together all of complex cognition, it's the practice of testing ideas for their quality. Some people continually monitor ideas for their quality; these people are good thinkers by habit and therefore good role models. How is it possible to establish higher-order thinking as a habit -- to build metacognition into our mental software as a "background application" that runs continuously? (p. 143)

    3. All the habits of mind we have considered -- reflection, mindfulness, thinking dispositions, and search -- are applicable across the various forms of complex cognition. These habits can make us more effective as we engage in problem solving, critical thinking, inductive reasoning, and creative expression. Habits of mind are about how we consistently -- how *habitually* -- we use the intellectual resources at our disposal. Here is yet another application to teaching and learning: Education entails, in part, the development of good habits of mind.