ABSTRACT

Mixed-Methods Study of the Impact of a Computational Thinking Course on Student Attitudes about Technology and Computation

William A. Booth, Ph.D.

Chairperson: Susan K. Johnsen, Ph.D.

The following dissertation reports on a mixed-methods, convergent parallel design research study of the impact of a computational thinking curriculum on attitudes towards computation and the use of technology. This study used a new curriculum in computational thinking recently developed through a National Science Foundation (NFS) Pathways to Revitalized Undergraduate Computer Education (CPATH) grant as the intervention. The purpose of this one-semester course is to introduce computational thinking to undergraduate students who are not computer science majors. This course is intended to engage a broad range of students, including those not ordinarily accustomed to using computation as a problem solving tool.

The treatment group was represented by twenty-two students randomly selected from an information technology course. The traditional curriculum for this information technology course was modified to include topics from the computational thinking curriculum. Topics included problem abstraction and decomposition, understanding

fundamental programming concepts, and appreciating the practical and theoretical limits of computation. Those students enrolled in the information technology course, not selected for the treatment group, used as the comparison group. All students in the study took a computer anxiety survey at the beginning and end of the semester. The participants also completed frequent surveys to report on their level of anxiety and insights into the efficacy of the instructional methods being used in the classroom. In addition to the survey data, the participants weekly lab reports were included in the qualitative data collected about the use of computational thinking strategies in problem solving. Finally, semi-structured interviews were conducted with each member of the treatment group to gain insight into how they used technology before receiving formal training in computational thinking.

The qualitative data were encoded and analyzed for evidence of the course's impact on the participants' attitudes towards and application of computational strategies in problem solving. This study found that when students participate in a computational thinking course there is in increase in the use of computational strategies in problem solving, an increase in positive affect towards computational thinking and a decrease in computer anxiety.

Mixed-Methods Study of the Impact of a Computational Thinking Course on Student Attitudes about Technology and Computation

by

William A. Booth, B.S., M.S.

A Dissertation

Approved by the Department of Educational Psychology

____
Marley W. Watkins, Ph.D., Chairperson

Submitted to the Graduate Faculty of
Baylor University in Partial Fulfillment of the
Requirements for the Degree
of
Doctor of Philosophy

Approved by the Dissertation Committee

____
Susan K. Johnsen, Ph.D., Chairperson

____
Janet H. Bagby, Ph.D.

____
Julie K. Ivey-Hatz, Ph.D.

____
William B. Poucher, Ph.D.

____
Tony L. Talbert, Ph.D.

Accepted by the Graduate School
May 2013

____
J. Larry Lyon, Ph.D., Dean

Page bearing signatures is kept on file in the Graduate School.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

This project could not have been completed without the support of my family, friends, and a dedicated, caring community of scholars. I would like to thank my mentor and committee chair Susan K. Johnsen. Susan, I will miss our weekly meetings. Your guidance and support throughout this process have help to shape me as both an academic and teacher.

I would also like to thank the Department of Educational Psychology. It is a blessing to be taught by a dedicated and talented faculty. An additional thanks goes out to the Department of Computer Science, it is a joy to work with this special group of friends.

Finally, I would like to thank my family. Tracy, your love and support mean the world to me. Katie and Rachel, I am so proud of you, you fill my heart with joy. Mom, you and Dad helped to shape me into the man I am today. You formed the foundation that has made the completion of this project possible.

DEDICATION

To Tracy

CHAPTER ONE

Introduction

Computational thinking (CT) has existed since the beginning of computer science. It was originally known as *algorithmic thinking* in the 1950s and 1960s (Denning, 2009). Computational thinking is the application of specific problem solving strategies used by computer scientists; it is a way humans solve problems, not computers (Wing, 2008). The constraints imposed by the digital computer is what makes the problem solving computational. Many problems are not easily solved using a computational framework. It would be difficult to think of a computational model that would facilitate painting a picture or getting a good night's sleep. However, if a problem can be abstracted into a computational framework, then a wide rage of previously unavailable strategies can be brought to bear on the problem solving process (NRC, 2009).

Using computation as a problem solving strategy may involve dividing a problem into its parts, and then solving each part separately or choosing an appropriate representation for a problem or modeling the relevant aspects of a problem to make it tractable. CT thus is using abstraction and decomposition when attacking a large complex task or designing a large complex system (Astrachan, 2009).

Computational thinking encompasses a rich set of skills including formulating problems, making abstractions, and phrasing solutions in ways that can be satisfied computationally. These skills range from algorithms and data structures to presentations and visualizations (Hambrusch, Hoffmann, Korb, Haugan, & Hosking, 2009). Although many of these skills are used in non-computational problem solving, the full extent of their power is only realized when the burden of computation is removed from the human and given to a computational tool such as a digital computer.

Another important computational skill is recursion. This is the idea that a problem can be solved by reducing it to a smaller or simpler instance of the original problem. Through a process of repeated reduction eventually the problem is reduced to a trivial state. The solution to the trivial instance is then used to construct a solution to the more complex original problem. The key point is that recursion is a common-place occurrence and should not be treated as an exotic construct in computation (Eisenberg, 2008).

Of all the skills associated with computational thinking, abstraction is perhaps the most difficult to master. Conceptualizing a problem through abstraction allows the problem solver to find links between problems and reuse solutions (Qualls & Sherrell, 2010). The interdisciplinary aspect of computational thinking allows it to alter the way many problems are solved, in a variety of fields (Curzon, Peckham, Taylor, Settle, & Roberts, 2009). The skill of problem abstraction can be directly applied in non-computational settings allowing the problem solver to see a problem through multiple perspectives.

Significance of the Study

The influence of computers in society for both professionals and students is of increasing importance and interest. Computer technology has become nearly ubiquitous in the last decade, but this technology holds little utility if people are not able to use it effectively. Wing suggests that "Computational thinking is a fundamental skill for everyone, not just for computer scientists." She defines computational thinking as "...solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science"(Wing, 2006, p. 33).

The number of students selecting computer science as a college major is declining in the United States. At the very same time the demand for computer scientists is increasing (Board, 2010). The Bureau of Labor and Statistics estimates that the number of computer science jobs in the United States will increase by 29.2% during the ten-year period from 2006 to 2016 (Bureau of Labor Statistics, 2007). Many of these jobs are in highly sensitive areas involving national security, defense and the financial/banking industry. These jobs cannot be filled by foreign nationals. The result is that many of these jobs are filled by individuals who are not fully qualified. The negative perception of computer science must be improved if sufficient numbers of students are to be recruited to computational sciences to meet this growing demand. One way to address this negative perception is to teach computational thinking as a problem solving strategy.

One indication of the importance of this way of thinking is that the National Science Foundation (NSF) has recently begun funding research that seeks to develop course work and methodologies for teaching computational thinking. In 2009, "the Directorate for Computer and Information Science and Engineering (CISE) of the National Science Foundation released a revised solicitation for the CISE Pathways to Revitalized Undergraduate Computing Education (CPATH) Program" (NSF, 2009, p. 4). In the revised solicitation all grant proposals must include computational thinking to be considered for funding.

In a technologically advanced society, citizens who lack a solid foundation in, and understanding of technology are at a disadvantage. They do not know what problems can be solved by computers and when and how technology can be appropriately applied to make their lives easier and increase the quality of the human experience (Wing, 2008).

Issues

Several key issues motivate the teaching of computational thinking and the study of the impact of this way of thinking. First, there is a need for participants in a technologically rich society to possess basic competency with computational tools. Next, computational thinking courses focused on programming do not address problem solving.

Finally, there is a need to correct a poor public perception of careers in computing. Each of these issues will be discussed below.

Computational Thinking has broadened the range of the problem solving skills needed to be a productive member of any modern society (Curzon et al., 2009). Students and practicing professionals in information-based fields, like science and engineering, require a basic competency with computational tools, but this ability is increasingly necessary for individuals in non-science fields. Because membership in any modern society requires the use of technology, an understanding of its capabilities and limitations is increasingly critical to the successful participation in everyday life. In some areas the divide between the creation of technology and its application has created a situation where particular individuals are unable to adequately use computational tools. It is not uncommon for individuals who use technology, even proficient users of technology, to be unfamiliar with best practices and even what sorts of computing tasks are possible (Dziuban, Moskal, & Hartman, 2005).

Moreover, formal instruction in problem solving has been shown to improve problem solving skills. In a recent study conducted by Hong Qin (2009) found that when computational thinking emphasized problem solving, students were able to recognize the common pattern of thinking processes among diverse problems and appreciated the value of thinking beyond solving a particular problem. They were able to generalize one problem into a class of problems with similar solution strategies. This was true even for students who previously seemed uncomfortable during one-on-one interaction with a computer (Qin, 2009b).

A key feature that must be addressed is separating the idea of computational thinking from the act of programming (Qualls & Sherrell, 2010). Programming is a process used by computer scientists and programmers to construct a set of instructions for a digital computer to solve a particular problem or class of problems. Thinking computationally is not simply programming; instead, it is a way of looking at problems through a filter designed to optimize the potential benefit of applying technology to problem solving. Thus, computational thinking is a key aspect of programming but is not unique to computer science (Denning, 2009). To provide the greatest impact and make computational thinking accessible to the maximum number of people computational thinking courses should not be programming centric. Introductory Information Technology courses for non-computer science majors are an ideal venue to allow students to explore abstraction, decomposition, problem solving, software engineering techniques, relational database design, programming concepts and constructs (Bryant et al., 2009).

Teaching computational thinking may also help to correct a poor public perception of careers in computing. Perceptions of professionals in the field of computing as socially marginalized individuals who spend the majority of there time isolated from human interaction while writing computer software are not accurate. This negative perception impacts the number of college students choosing computing as a major (Hambrusch et al., 2009). Computing professionals spend the majority of their time interacting with others and looking for ways to abstract problems that allow them to be solved with a computational strategy.

Theoretical Framework

This convergent mixed method research project takes a pragmatic ontological world view. This ontological perspective acknowledges the existence of a single reality with multiple interpretations of that reality. Although there is a single reality all participants in that reality experience it differently, from a deeply personal perspective (Creswell & Plano-Clark, 2011).

Because this research seeks to investigate factors influencing how people learn and what impact learning has on the learner's perception, key features of constructivism interpreted through the lens of Kolb's Experiential learning theory (ELT) will be used as the theoretical framework for this study. The key attributes of constructivism are (Lainema, 2009):

(1) Learning is an active process of constructing rather than acquiring knowledge

(2) Instruction is a process of supporting that construction rather than communicating knowledge.

The constructivist world view holds that learning is an active, contextualized, constructive process. The learner constructs meaning from information gathered through their senses within some context. People actively construct their own subjective representations of objective reality (Lainema, 2009).

Experiential Learning Theory (ELT) differentiates itself from other cognitive learning theories through its emphasis on experience . ELT traces its roots to the work of Dewey, Lewin, and Piaget (D. A. Kolb, Boyatzis, & Mainemelis, 2000). "Taken together, Dewey's philosophical pragmatism, Lewin's social psychology, and Piaget's cognitive-developmental genetic epistemology form a unique perspective on learning and development" (D. Kolb, 1984). The ELT holds that learning takes four distinct forms. Two modes of grasping experience, Concrete Experience (CE) and Abstract Conceptualization (AC), and two modes of transforming experience Reflective Observation (RO) and Active Experimentation (AE) (D. A. Kolb et al., 2000). Kolb (2000) gives the following description of this model:

> The ELT model portrays two dialectically related modes of grasping experience – Concrete Experience (CE) and Abstract Conceptualization (AC) – and two dialectically related modes of transforming experience – Reflective Observation (RO) and Active Experimentation (AE). According to the four-stage learning cycle depicted in Figure 1, immediate or concrete experiences are the basis for observations and reflections. These reflections are assimilated and distilled into abstract concepts from which new implications for action can be drawn. These implications can be actively tested and serve as guides in creating new experiences (D. A. Kolb et al., 2000).

Concrete Experience is learning through first hand experience. In the first form of learning the learners senses are used to perceive the world and these sensory inputs are then used to construct meaning. The second modality of learning is analytical. The learner uses symbolic representation or abstract conceptualization as a model

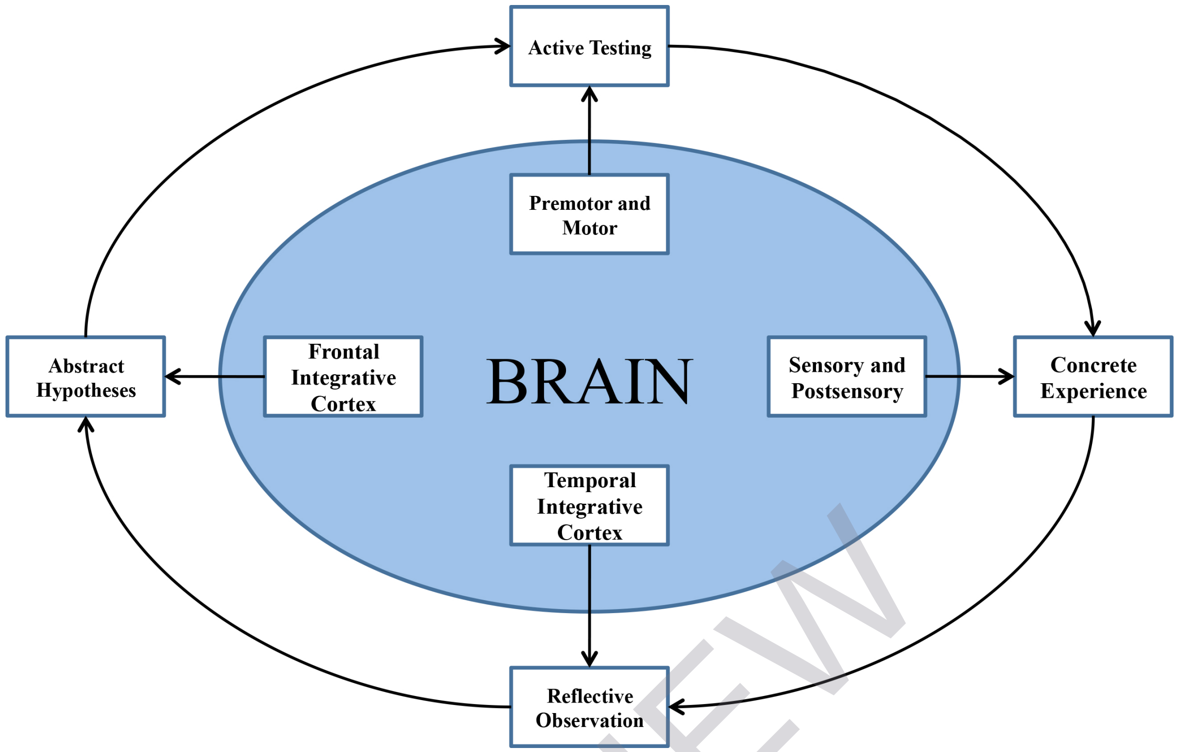

Figure 1.1. Experiential Learning Theory (A. Y. Kolb & Kolb, 2005, 195)

to thinking about, analyzing, or systematically planning, rather than using physical sensation. These two modalities of learning are mutually exclusive and require the learner to choose one or the other (D. A. Kolb et al., 2000). In learning a new task some learners prefer to watch a mentor or expert and carefully analyze the new information to construct new meaning before attempting to apply these new skills. Other learners prefer to discover for themselves how things work. It is part of the human condition that causes these two modalities of learning to be mutually exclusive. It is almost impossible to simultaneously read a driving manual while actually driving a car (D. A. Kolb et al., 2000). "We resolve the conflict between concrete or abstract and between active or reflective in some patterned, characteristic ways. "(D. A. Kolb et al., 2000, p. 4).

The four-stage learning cycle described above was used to explain how the participants in this study are learning to use computational thinking as a problem solving strategy. First, the learners were presented with a problem that could be solved computationally. The instructor then demonstrated how to construct a computational solution to this problem. This initial problem was followed by additional similar problems that allowed the learners to practice the computation problem solving strategy.

Each unit in the computational thinking curriculum provided the learner with an example of a computational solution to a given problem type. The presentation of each example problem was the concrete experience (CE) that provided an opportunity for reflective observation (RO). Once these RO are assimilated into an abstract concept the learners formed an abstract hypothesis (AH) that was used during the active experimentation phase of each module. The AE phase of each instructional module was conducted during the weekly labs where the learners were asked to apply the new problem solving strategy to a problem similar to the one presented by the instructor.

Rational for the Study

It is part of the human condition to solve problems. A human being's ability to problem solve is a critical aspect of what it means to be human. Professors of computer science should teach a course called "Ways to Think Like a Computer Scientist" to college freshmen, making it available to non-majors, not just to computer science majors. These courses should help the students answer questions like. What is computable? What is intelligence? What is information? How can we build complex systems simply? Computational thinking is a grand vision to guide computer science educators, researchers, and practitioners as they seek to change society's perceptions about problem-solving and about the field of computer science (Wing, 2006). In a technology rich society, computational thinking is a powerful problem solving strategy. With the rich collection of computational tools available to all members of our modern society there is a need to educate the general public about what is possible with the help of technology and what is not.

The goal of this study is to determine if formal instruction in computational thinking can both increase the use of this problem solving strategy and positively impact the learners' attitudes towards the use of technology in problem solving. Qualls (2010) notes that educators must not wait until high school or college to teach these ideas, computational thinking should be taught in elementary school to help ensure that all citizens are well prepared to live in our modern society (Qualls & Sherrell, 2010).

There are few studies represented in the research literature that address the topic of computational thinking. The field has defined what is, and is not, computational thinking (Wing, 2006). Researchers are just now beginning to publish results of computational thinking studies with early articles in the area focusing on anecdotal results and reports of best practices.

Results from this study will help to inform educators and students about the importance of computational thinking. Teachers, students and the general public need to know what computational thinking is, what types of problems can be solved computationally, and the capabilities of modern computational tools.

Research Questions

An important initial step in any research is the careful construction of the research questions (Yin, 2008). Both qualitative and quantitative questions were asked to investigate different aspects of the impact of teaching a course in computational thinking (CT).

The quantitative strand investigated three effects of formal training in computational thinking. Research question one (RQ1) looked at what effect CT training has on the frequency of CT as a problem solving strategy. Research question two (RQ2) looked at how formal training in CT impacts the quality of the computational solutions. Research question three (RQ3) investigated how formal training in CT effects the learners' level of computer anxiety.

The qualitative strand also investigated three effects of formal training in CT. Research question four (RQ4) investigated the effect of training in CT on the learners' attitudes about CT. Research question five (RQ5) looked for factors that impact computer anxiety when learners are taught CT. Research question six (RQ6) investigated how formal training in CT changes they war the learners solve computational problems.

The final two questions, research question seven (RQ7) and research question eight (RQ8), were answered by combining the evidence from the qualitative and quantitative strands into a metanarrative. RQ7 addressed the effect of CT on attitudes about the use of computation in problem solving and, RQ8 investigated the impact of training in CT on the learner's understanding of solving computational problems.

The following are research questions for each category in this dissertation.

Quantitative Research Questions

RQ1 Do students who participate in formal training in computational thinking increase the frequency of the use of computational strategies in problem solving when compared to students who do not?

RQ2 Do students who participate in formal training in computational thinking produce quality solutions to computational problems?

RQ3 Do students who participate in formal training in computational thinking decrease their computer anxiety when compared to students who do not?

Qualitative Research Questions

RQ4 In what ways does formal training in computational thinking change attitudes of students who participate in a course in computational thinking?

RQ5 In what ways does formal training in computational thinking change computer anxiety of students when compared to students who do not?

RQ6 In what ways does formal training in computational thinking change the computational strategies of students?

Mixed Methods Research Questions

RQ7 How does formal training in computational thinking impact the learners' attitudes towards the use of computational strategies in problem solving?

RQ8 How does formal training in computational thinking impact the learners' understanding of computation and its application to problem solving?

Each question was crafted to guide this study in determining the impact of teaching computational thinking on the attitudes and capabilities of the students in the treatment group. Different dimensions of each question were investigated through the qualitative and quantitative strands of this study.

Conclusion

Computer technology has become nearly ubiquitous in our modern society, but this technology holds little utility if people are not able to use it effectively. The influence of computers in society for both professionals and students are of increasing importance and interest. Computational Thinking has broadened the range of the problem solving skills needed to be a productive member of any modern society (Curzon et al., 2009). Despite the increase in availability and use of computational tools, the number of students selecting computer science as a college major is declining in the United States (Board, 2010). This study seeks to determine the impact of formal training in computational thinking has on the attitudes of learners towards computation and the use of technology in problem solving. The study also documented the impact of formal training in computational thinking on the frequency of computational thinking as a problem solving strategy.