Personality and Programming

Amy B. Woszczynski
Tracy C. Guthrie
Sherri Shade
Computer Science and Information Systems Department
Kennesaw State University
Kennesaw, GA 30144 USA
awoszczy@kennesaw.edu tguthri@bellsouth.net sshade@kennesaw.edu

ABSTRACT

Information systems students continue to struggle to successfully complete computer programming classes. Learning how to program is difficult, and failure and attrition rates in college level programming classes remain at an unacceptably high rate. Since many IS students take a programming course as part of their program of study, IS educators should better understand why IS students tend to achieve low success rates in programming courses and what can be done to improve success rates. Little research to date has addressed potential reasons for student failure in programming principles courses. Many educators simply assume that high failure rates are acceptable – that computer programming is difficult and some students simply will not succeed. Some researchers have studied personality as a predictor of success in computer programming courses. However, no studies have attempted to gather cognitive profiles and match performance to profile type exhibited. In our study, we identified the primary cognitive profile in a sample of beginning programming students in a southeastern university and matched profile to final average in programming principles I. Intuitive thinkers tended to perform better in programming principles I than sensor feelers. We found no other differences in performance between profile types. We recommend instructional strategies that may be used to reach fully motivated and intellectually capable sensor feelers, while not detracting from the learning experience of the other profiles.

Keywords: programming principles, cognitive profiles, personality, CS1

1. INTRODUCTION

For years, scientists and educators have studied the psychological makeup of people and have been curious about the ways that people learn and retain information. The study or science of cognitive profiles has been dissected in many different ways through the years with some of the earliest work performed by a Swiss scientist, Dr. Carl Jung (Campbell, 1971). Jung's studies attempted to group personality profiles into structures or combinations of one selection from four pairs of possibilities. These early studies led to the use of Jung's profiles in the Myers-Briggs Type Inventory (Corns, 1998; Ring, 1998), which sometimes is used to study how people interact in society.

This paper extends Jung's work by applying personality research and in particular, cognitive profiles, to an analysis of a typical gatekeeper course taken by IS students: programming principles I – often referred to as simply CS1. Here, we use CS1 as a surrogate for CS111, Introduction to Programming, in ACM's Computing Curricula 2001 (http://www.acm.org/sigcse/cc2001). The ACM guidelines describe CS111 as covering topics such as:

> ...standard programming constructs, problem-solving strategies, the concept of an algorithm, and fundamental data structures (strings, arrays, and records) along with an introduction to machine representation, graphics, and networking.

Across universities and curricula, CS1 has a notoriously low rate of success – defined here as earning an A, B, or C in the course. Personality research offers great potential for giving educators and researchers more information on why so many students fail to succeed in CS1. Krause (2000) demonstrated that students may learn in different ways based on their personality profile, and our research identifies potential instructional strategies to reach consistently underperforming groups. By having students take an online profile developed by Krause, we hope to better direct student activities to achieve higher success rates in CS1. After students take the profile, the provision of appropriate study techniques offers an enhanced chance of successfully learning and retaining material.

For this project, CS1 students took either a paper and pencil or online version of the Cognitive Profile Inventory (CPI) (Krause, 2000). Then we tracked student performance in the course by personality type to identify groups that might need different pedagogy and instructional strategies to reach their full potential for the successful completion of CS1. By matching student information to the personality profile type, we can measure success based on personality characteristics and provide appropriate instructional strategies to help diverse personality types succeed in CS1. In this research, we recommend corresponding instructional strategies to reach groups that perform poorly. We believe that appropriate instructor pedagogy and instructional strategies can improve the success rate of intellectually capable and properly motivated students. The next section provides a brief overview of previous research into computer programming, personality, and cognitive profiles.

2. THEORETICAL BACKGROUND

2.1 Difficulty in Computer Programming

Computer programming includes many aspects of learning; it requires the prospective programmer to analyze problems, implement solutions in a programming language, execute the solution in a computer operating system, track, follow, and debug code if necessary, and make enhancements to the program to further the effectiveness of the solution. Another way to understand the difficulty of learning how to write computer programs is to imagine receiving a document written in an unfamiliar foreign language with the assignment to read, process, and assign a solution for the stated problem.

Students in computer programming curriculums have traditionally struggled with one or more of the concepts required for success in this field. The most evident statistics of this are from the beginning programming classes where failure and course withdrawal rates often exceed 50% or more. In a recent study, researchers studied the pass, fail, and withdrawal figures for the CS1 classes where successful completion of the course requires a grade of C or better (Beise, Myers, VanBrackle, and Chevli-Saroq, 2003). Beise and colleagues found that the overall probability of passing CS1 the first time was 40% across all majors, with an initial failure rate of 19.5%, and a withdrawal rate of 40.5%. For IS students, the passing rate was even more dismal, with 32% passing the course, 16% failing the course and 52% withdrawing from the course. Clearly, IS educators need to be concerned about the poor performance of their student majors in this gatekeeper course.

On initial observation of these statistics, one could argue that regardless of the technical acumen that a student possesses, the failure rate in all fields is very high. Is this rate caused by the degree of difficulty involved in the course? Is it due to poor study habits on the student's part? Does the low student success rate simply highlight that the learning of computer programming is unique and one to which students have limited exposure until they reach the college level of instruction? Alternatively, do these findings and other similar studies point to the fact that all people learn differently, and respond to differing academic and environmental stimuli when learning in any specialized discipline? This study attempts to address some of these issues, after first discussing personality and cognitive profiles.

2.2 Personality

In Jung's seminal work on personality profiles entitled Psychological Types (Jung, 1971), he sought to identify people as introverts or extroverts, where introverts relate more to interest in a subject, and extroverts focus more specifically on objects in their environment. That is, extroverts are more outgoing and sociable, while introverts may be withdrawn and shy. Later work has built upon Jung's pioneering efforts, resulting in a plethora of studies in organizational behavior and psychology (Campbell, 1971).

After identifying subjects as either introverts or extroverts, Jung then sought to break a person's psyche down into sections that are more detailed. The identified functions or thought processes are: thinking and feeling, which Jung argued were rational thought processes; and sensation and intuition that he felt were irrational thought processes.

Myers-Briggs took the initial findings of Jung and attempted to break the profiles down even further into sixteen different personality types (Myers, 1962). One of the main differences between the Jung study and the Myers-Briggs study is that Myers-Briggs did not attempt to identify and group a person into one category, but felt that a person – while having a preference on each side of the four main categories – could cross the boundaries in a given situation. This corresponds to the widely regarded interactionist viewpoint, in which inherent personality characteristics and the situation in which a person is placed both play a role in how that person behaves (Bowers, 1973; Caspi, 1987; Funder and Dobroth, 1987; Magnusson, 1981). Myers-Briggs has found high acceptance rates in many different areas, and is used quite regularly for businesses seeking to provide more in depth analysis of employment candidates (Harvey, Murry, and Markham, 1995).

Several studies have examined the role of personality in information technology related courses, with somewhat conflicting results. For example, Kim and Schniederjans (2004) used the Wonderlic Personality Characteristics Inventory and found that personality indicated course grade for juniors taking a Web-based introduction to MIS course. In addition, several IS studies have used Myers-Briggs or similar measures of cognitive style to determine personality and its relationship to course performance or potential educational strategies. For example, Carland and Carland (1990) studied the personality type of computer information systems majors and made pedagogical recommendations to help reach all types of learners. Further, van Merrienboer (1990) researched students taking computer programming courses and described instructional strategies to reach students with different cognitive styles. Finally, Capretz (2002) used results from Myers-Briggs testing to develop instructional strategies in a software engineering course to reach students with different personality types.

However, even with multiple studies on personality and computer programming, the results to date have been somewhat fragmented, with some studies showing that personality is related to course grade in CS1 and other studies showing no relationship between the two. Bishop-Clark (1995) recommended further research to clarify the conflicting results, and this paper provides a new perspective using the CPI. Moreover, most previous research has studied CS and engineering students, with little attention paid to IS students. Our research includes both CS and IS majors, which may in particular help IS educators better understand the factors that are related to success for IS students taking CS1.

Few would argue with the contention that some people exhibit artistic characteristics, some people have scientific tendencies, some people pursue more theoretical outlets, and others naturally succeed in mathematical endeavors. To a certain degree people's strengths cross these boundaries when they are skilled in multiple areas, but people always seem to have a weaker area of natural interest and innate ability. Most people find some areas very challenging with other areas seeming extremely easy. One method of addressing personality as it relates to improved study habits and greater success in the classroom is through the use of cognitive profiles, as described in the next section.

2.3 Cognitive Profiles

People come in various shapes, sizes, and colors, and researchers often study a variety of personality aspects to understand how people learn and absorb information. People react differently to all sorts of scenarios when they are attempting to learn or to study information. Some students learn best with no distractions, but others respond better when there is music blaring and conversations about other topics taking place. Some students respond better to interactive learning, while others prefer to memorize information on their own. Some students prefer certain test types such as multiple choice rather than intuitive testing structures such as those involved in essay or short answer tests. Whatever their mental aptitudes, there are identified profile types that generally respond to certain stimuli when attempting to learn or to obtain new knowledge.

The book How We Learn and Why We Don't (Krause, 2000) builds upon the work of Jung and colleagues and attempts to identify four distinct learning types using a word identification test that also helps to determine the correct study skills for the established dominant cognitive profile. The test is simple in nature and involves nothing more than the participants selecting one word from sixty pairs of words and selecting which word better describes them. The numerical results from the test produce a map or graph demonstrating the degree to which the student's cognitive results fall into the different cognitive profiles.

Krause (2000) divides learners into four cognitive profiles, using two letter acronyms that include ST for Sensor Thinker, SF for Sensor Feeler, NT for Intuitive Thinker, and NF for Intuitive Feeler. People who are sensors tend to learn or gain knowledge through the use of their senses, while intuitive learners use visual learning methods to recall information. Thinkers like to have concrete evidence for decisions, while feelers tend to make decisions based on emotion or morality.

Krause goes on to describe ST students as those who prefer to study by themselves with little distraction, while students with the SF profile group use structured thought processes and learn through repetition and breaking problems into steps or milestones. In contrast, NF students do well when they are allowed to build concepts from nothing and given freedom to try different ideas that may expand upon existing theories. The NT profile, on the other hand, tends to use pictures to enhance learning. Based on the ability of NTs to see something before it exists physically, we believe that they will be able to visually see how a program should be structured before it is actually written. This awareness of how the final product will look should help the NTs to plan and accurately develop a program and then code it properly.

Krause's Cognitive Profile Inventory (CPI) classifies people on the areas from Myers-Briggs that have shown close relationships to problem solving ability: sensor/intuitive and thinking/feeling (Bishop-Clark & Wheeler, 1994). Since problem solving plays an important role in writing computer programs, we believe that the CPI provides an appropriate personality test for measuring performance in CS1. Moreover, some researchers have called for using alternative personality measures in addition to Myers-Briggs in information systems research (Lampe, 2004), and we believe that the CPI provides a valid and shorter alternative to Myers-Briggs.

Each profile learns, processes, and retains information using different skills and innate abilities. Student cognitive profiles previously have demonstrated the ability to determine how students will comprehend information in the most efficient and beneficial way. Because each profile is different in varying degrees from the other identified profiles, and since some profiles are naturally more adept at technical acumen than others, we predict:

Hypothesis 1: Students with diverse dominant cognitive profiles will perform differently in CS1.

Feeling profiles tend to react to problem solving based on their feelings, emotions, or through the use of sensory perception. On the other hand, thinking profiles tend to gather and digest all of the pertinent information before seeking a relevant solution. Computer programming success generally requires the ability to collect and interpret information relating to a problem, followed by the application of this information to a corrective action. Further, as Bishop-Clark and Wheeler (1994) note, thinking students may enjoy and complete computer programming courses more than their feeling counterparts. Moreover, thinking students are also more likely to experiment with software (Jones, 1994). Based on the fact that thinking profiles are more inclined to process information in a manner conducive to successful writing of computer programs, we predict:

Hypothesis 2: Students with Thinker (NT and ST) profiles will have higher success rates in CS1 than Feeler (NF and SF) profiles.

Moreover, researchers have also studied the relationship between success in programming courses and gender. Although result have been mixed, with some studies finding a difference between women and men, Beise and colleagues (2003) recently found that there were no differences in success based on the sex of the student. Therefore, we propose:

Hypothesis 3: Regardless of profile, women and men will not have different success rates in CS1.

3. METHODOLOGY

In this study, we determined the dominant cognitive profile of programming principles students at a southeastern regional university. Participants received either: 1) a free online cognitive profile analysis and feedback on the identified study skills that have proven effective for their dominant profile type in past cognitive analyses or 2) a free copy of the book How We Learn and Why We Don't, by Lois Krause (2000) and accompanying survey. Both surveys were identical. We then tracked these students through the assignment of final grades for CS1 to determine if success rates increased with student awareness of cognitive profile and use of recommended study tools.

4. PARTICIPANTS

A sample of 247 students completed the cognitive profile exercise, with 49 of them taking the on-line survey and 198 completing the survey by hand. Students taking the paper-and-pencil survey were required to complete the exercise for class credit, but they were not required to have their results included in the study. Only five of the paper-and-pencil survey respondents elected not to have their results included in the study. In addition, six of the remaining students had multiple dominant profiles and were thus dropped from the analysis, leaving a final usable sample size of 236 students. Of those, 70 of the respondents were female and 166 male. The sample included a variety of majors, with computer science and information systems accounting for almost 75% of the respondents, as shown in Table 1.

5. RESULTS

We completed an analysis of variance (ANOVA) to test differences between groups. Hypothesis 1 was supported: Students with diverse dominant cognitive profiles performed differently in CS1 ($F=2.83$ and $p = 0.0394$). Students with NT profiles achieved the highest overall final averages, with an average grade of 2.92 on a 4.00 scale, as shown in Table 2. Those with NF profiles achieved average grades of 2.64, while ST profiles averaged 2.60. SF profiles scored lowest, with an average grade of 2.11 out of 4.00.

To test Hypothesis 2: Students with Thinker (NT and ST) profiles will have higher success rates in CS1 than Feeler (NF and SF) profiles, we used several tests of differences between means. Tukey's Studentized Range test, t-tests, Duncan's multiple range test, Scheffe's test, and Student Newman-Keuls tests all showed significant differences between NT and SF cognitive profiles at the 0.05 level. That is, NT profiles scored significantly higher than SF profiles. None of the tests revealed any differences between other profile pairs. Therefore, Hypothesis 2 was partially supported. Intuitive Thinkers (NT) did score significantly higher as a group than Sensor Feelers (SF). However, we had also predicted that NTs would score significantly higher than Intuitive Feelers (NF), and that result was not supported by any of the pairwise comparison tests. Further, Sensor Thinkers (ST) did not score significantly higher than either SFs or NFs, as we had predicted.

As expected, there were no significant differences in performance between men and women, regardless of profile type ($F=0.08$, $p=0.7776$). Men averaged 2.66 out of 4.00, while women averaged 2.60 out of 4.00, as shown in Table 3. Therefore, hypothesis 3 was supported.

Major	# of Students
Accounting	1
Art	1
Biology	1
Business/Management	3
Chemistry	1
Communications	1
Computer Science	104
Education	1
Finance	2
Information Systems	70
Joint Enrollment	1
Honors	
Mathematics/Math Ed	8
Graduate Majors	8
Nursing	1
Political Science	1
Psychology	2
Transient	6
Undecided	24
TOTAL	236
Sex	**# of Students**
Female	70
Male	166

Table 1. Demographic Characteristics of Sample

Dominant Profile	Final Average	Std Deviation	n
NT	2.92	1.47	77
NF	2.64	1.37	50
ST	2.60	1.33	73
SF	2.11	1.35	36
TOTALS	2.64	1.40	236

Table 2. Dominant Cognitive Profiles and Final Average (out of 4.00 scale) in CS1

Gender	Final Average	Std Deviation	n
Men	2.66	1.37	166
Women	2.60	1.48	70
TOTALS	2.64	1.40	236

Table 3. Gender and Final Average (out of 4.00 scale) in CS1

6. DISCUSSION AND LIMITATIONS

6.1 Potential Instructional Strategies

In general, our results seem to indicate that with the exception of the SF cognitive profile, all other profiles exhibit similarly positive outcomes in CS1. Therefore, the focus of our efforts should concentrate on bringing the SF profiles up to the level of the other profile groups. Since SF students tend to learn best through repetition and by breaking problems into steps or milestones, these students might benefit from starting with very small steps in constructing a program. After mastering one concept or one sequence of instructions, they can add another piece. These learners should respond most favorably to a learning environment where they can receive credit for portions of programs or for concepts learned. The movement to object-oriented programming (OOP) could provide SF students with the ability to develop these smaller modules of code as building blocks to solve larger or more complex problems. OOP could appeal to the SF profile and provide a more enriching CS1 experience for the learner who prefers to build and test solutions to problems in smaller steps. Moreover, the move to Web based and visual programming environments could also appeal to the SF learner as these alternatives build upon appearance that is artistic in nature, use small programming modules to meet specific needs, and have dynamic presentations to serve multiple users.

Since SF learners attempt to relate the information to real life objects with which they are familiar, the use of more real-world problems in CS1 should also benefit them. Further, most educators would agree that using real-world problems brings relevance to the classroom for all students, regardless of personality profile. Therefore, integrating real-world problems into the classroom should make the course more useful and relevant to a wide variety of students.

In addition, since SF profiles tend to perform well when studying and working in pairs or groups, we see great potential for the use of innovative ideas in the classroom, such as pair programming. Studies have noted the potential for pair programmers to produce better code than individuals (Nagappan, Williams, Ferzli, Wiebe, Yang, Miller, and Balik, 2003), and students also perceive benefits when participating in the pair programming experience (VanDeGrift, 2004). In fact, Bishop-Clark and Wheeler (1994) noted that sensors had a higher average on programming assignments than their intuitive peers. Since all of the assignments in Bishop-Clark and Wheeler's study were completed in groups, we believe this result supports our recommendation to use more group activities to appeal to sensors. Moreover, since many programmers will find themselves working in teams when they enter the workforce, we contend that paired programming should benefit all types of learners. Further, we should encourage the students with SF profiles to find partners and study in groups to maximize the learning experience. Pairing SFs with other SFs may provide an enhanced outcome for multiple students. The SF learner might find it advantageous to attend outside laboratory environments and/or smaller, recitation-like reviews where they can receive additional practice – that is, repetition – in programming concepts. These environments give the SF learner the ability to ask questions in a less-threatening environment while also giving opportunities to practice the concepts covered in the classroom.

We believe that offering all students the opportunity to learn about their personality may help the students study and perform more effectively. In fact, studies on pair programming found that students were sometimes teamed with incompatible personality types and complained about working with partners that had different personalities (Nagappan et al., 2003; VanDeGrift, 2004). In addition, Gorla and Lam (2004) extended the analysis to software projects and described the importance of understanding personality when forming teams. Moreover, researchers have noted the importance of students understanding their own personality strengths and weaknesses (Carland & Carland, 1990). By providing students and/or instructors with personality assessments, pair programming may become an even more effective option to team compatible students with each other when programming. Many universities offer free or reduced-price personality testing and career counseling, and students could be encouraged to use these services. Integrating an out of class on-line exercise that provides students with the opportunity to learn about themselves and how best to study to achieve success is also an option. None of these options should detract from the performance of the higher-achieving groups, and each of the options mentioned has minimal or no costs. Indeed, by offering a more welcoming environment to a diverse group of students, information systems and computer science students should benefit through the advent of new ideas and new types of individuals in the classroom and ultimately, in the working environment.

6.2 Limitations

Although we had a large usable sample size of 236 students, we had small numbers of the SF profile, with only 36 students, or about 15% of the sample. NT profiles represented the largest percentage of students, with over 30% of the sample. These results were not unexpected, however, since information systems and computer science students, who made up a large portion of our sample, often fall into the NT profile. Further, almost 75% of our respondents were either computer science or information systems majors. Future studies could sample a more diverse population to see if the results hold true with varying numbers of different profile types and across a variety of majors.

In addition, we analyzed only the correlation between cognitive profile and final average in CS1. Other variables may also play a role in performance in CS1. In this study, we provide a succinct analysis of the potential role of personality and its relationship to success in CS1. Future researchers should expand the model to include other relevant variables and determine which network of variables provides the best prediction of success in CS1.

Moreover, although our results indicated no differences in performance based on gender, less than 30% of the participants in our study were female. Therefore, we recommend that future studies use a sample that includes a larger number of women.

Further, our study used self-reports to gather data on respondents. Since multiple studies have noted the inherent limitations of self-reports (Woszczynski and Whitman, 2004), we recommend gathering data using multiple methods to reduce bias caused by using a common method. Moreover, using longitudinal studies to track performance over time throughout the college career would help to overcome limitations of self-reports.

6.3 Future Research

The current study offers the potential for several follow-up studies. For example, previous studies have indicated that women tend to have better learning experiences when they have a teacher that is also a woman (Rosser, 1993). Since women have better learning outcomes when they have a teacher that is similar to them, we could theorize that students will have better learning outcomes when they have a teacher who is similar to them in personality profile. Specifically, students who have the same profile as their instructor may be more successful than students with different profiles – that is, students will perform better in classes where the instructor and student have the same personality profiles. Our study only had five instructors who participated, and only three of those instructors completed the CPI. Therefore, we were unable to statistically test for instructor profile interaction with student profile. However, we believe that instructors would also benefit by learning about their own personality. We believe this area of research offers great potential for designing a classroom that offers all capable learners the ability to succeed, no matter their profile or the profile of their instructor.

7. CONCLUSIONS

Our study does not attempt to find a way to reach all students who struggle with programming principles. We fully agree that some students are not meant to major in information-technology related fields. However, if we can modify the pedagogy and curriculum delivery mechanism for programming principles so that students who have the desire and intellectual capability to succeed are able to succeed, then we will increase the diversity in the field over time. By integrating some of the instructional strategies recommended above and by further analyzing the link between instructor personality and student performance, we believe that educators will be able to reach those students who are fully motivated and capable of succeeding in programming principles, but who struggle with the method in which programming principles is presented. Will there still be a high percentage of students who fail to successfully complete programming principles? We believe the answer is yes. But will we reach a more diverse group and one that can further enrich the information technology field? We believe the answer to that question is also yes, and we hope that educators will implement, test and report upon some of the instructional strategies that we have recommended.

8. REFERENCES

Beise, C., Myers, M., VanBrackle, L, and Chevli-Saroq, N. "An Examination of Age, Race, and Sex as Predictors of Success in the First Programming Course." Journal of Informatics Education and Research, Vol. 5, No. 1, 2003, pp. 51-64.

Bishop-Clark, C., and Wheeler, D. "The Myers-Briggs Personality Type and its Relationship to Computer Programming." Journal of Research on Computing in Education, Vol. 26, No. 3, Spring 1994, pp. 358-370.

Bishop-Clark, C. "Cognitive Style and its Effect on the Stages of Programming." Journal of Research on Computing in Education, Vol. 27, No. 4, Summer 1995, pp. 373-386.

Bowers, K.S. "Situationism in Psychology: An Analysis and Critique." Psychological Review, Vol. 80, 1973, pp. 307-336.

Campbell, J. "Psychological Types." in The Portable Jung, R.F.C. Hull (ed.), Viking Press, New York, 1971, pp. 178-269.

Capretz, L. F. "Implications of MBTI in Software Engineering Education." inroads, SIGCSE Bulletin, Vol. 34, No. 4, December 2002, pp. 134-137.

Carland, JoAnn C., and Carland, James W. "Cognitive Styles and the Education of Computer Information Systems Students." Journal of Research on Computing in Education, Vol. 23, No. 1, Fall 1990, pp. 114-126.

Caspi, A. "Personality in the Life Course." Journal of Personality and Social Psychology, Vol. 53, 1987, pp. 1203-1213.

Corns, S. "Workshop Shows What Makes People Tick." Arizona Daily Wildcat, December 8, 1998. Retrieved on October 20, 2002 from http://wildcat.arizona.edu/papers/92/73/13_1_m.html

Felder, R. "Learning and Teaching Styles in Engineering Education." 1988, Retrieved October 21, 2002 from http://www.ncsu.edu/felder-public/Papers/LS-1988.pdf

Funder, D., and Dobroth, J. "Differences between Traits: Properties Associated with Interjudge Agreement." Journal of Personality and Social Psychology, Vol. 52, 1987, pp. 409-418.

Gorla, Narasimhaiah, and Lam, Yan Wah "Who Should Work with Whom? Building Effective Software Project Teams." Communications of the ACM, Vol. 47, No. 6, June 2004, pp. 79-82.

Harvey, R., Murry, W., and Markham, S. "A 'Big Five' Scoring System for the Myers-Briggs Type Indicator." May 1995, Retrieved November 10, 2002 from http://harvey.psyc.vt.edu/Documents/Personality/bigfive.doc

Jones, W. Paul "Computer Use and Cognitive Style." Journal of Research on Computing in Education, Vol. 26, No. 4, Summer 1994, pp. 514-522.

Jung, C. G. Psychological Types (Collected Works of C. G. Jung, Vol. 6). Princeton University Press, 1971.

Kim, Eyong B., and Schniederjans, Marc J. "The Role of Personality in Web-Based Distance Education Courses." Communications of the ACM, Vol. 47, No. 3, March 2004, pp. 95-98.

Krause, L. How We Learn and Why We Don't, Thomson Learning, Cincinnati, OH, 2000.

Lampe, James C. "Alternative Personality Measurements: Commentary on Accounting Information Systems Research Opportunities Using Personality Type Theory and the Myers-Briggs Type Indicator." Journal of Information Systems, Vol. 18, No. 1, Spring 2004, pp. 21-34.

Magnusson, D. (Ed.) Toward a Psychology of Situations: An Interactional Perspective, Erlbaum, Hillsdale, NJ, 1981.

Myers, I. B. The Myers-Briggs Type Indicator, Educational Testing Service, Princeton, NJ, 1962.

Nagappan, Nachiappan, Williams, Laurie, Ferzli, Miriam, Wiebe, Eric, Yang, Kai, Miller, Carol, and Balik, Suzanne "Improving the CS1 Experience with Pair Programming." SIGCSE '03, February 19-23, 2003, Reno, NV, pp. 359-362.

Ring, B. "Myers-Briggs Type Indicator: A Research Report." Personality Plus, 1998, Retrieved October 19, 2002 from http://members.tripod.com/~PersonalityInstitute/Myers-BriggsTypeIndicator.htm

Ross, K. Carl Gustav Jung (1875 –1961), 1997, Retrieved on September 23, 2002 from http://www.friesian.com/jung.htm

Ross, K. Psychological Types (After C.G. Jung and the Briggs – Myers Typology), Retrieved September 23, 2002 from http://www.friesian.com/types.htm

Rosser, S. V. (1993). "Female Friendly Science: Including Women in Curricular Content and Pedagogy in Science." Journal of General Education, Vol. 42, No. 3, 1993, pp. 191-220.

van Merrienboer, Jeroen J. G. "Instructional Strategies for Teaching Computer Programming Interactions with the Cognitive Style Reflection-Impulsivity." Journal of Research on Computing in Education, Vol. 23, No. 1, Fall 1990, pp. 45-53.

VanDeGrift, Tammy "Coupling Pair Programming and Writing: Learning about Students' Perceptions and Processes." SIGCSE '04, March 3-7, 2004, Norfolk, VA, pp. 2-6.

Webster, R. "Interfaces for E-learning: Cognitive Styles and Software Agents for Web-based Learning Support." 2001, Retrieved September 15, 2002 from http://www.medfac.unimelb.edu.au/ascilite2001/pdf/papers/websterr.pdf

Webster, R. "Learning Styles and Design: The Use of ASSIST for Reflection and Assessment." 2002, Retrieved September 19, 2002 from http://www.ecu.edu.au/conferences/herdsa/papers/ref/pdf/Webster.pdf

Woszczynski, A., and Whitman, M. (2004). "Chapter V. The Problem of Common Method Variance in IS Research." in The Handbook of Information Systems Research, M. Whitman and A. Woszczynski (eds.), Idea Group Publishing, Hershey, PA, 2004, pp. 66-77.

AUTHOR BIOGRAPHIES

Amy B. Woszczynski is an Associate Professor of Information Systems and Director of the MSIS Program at Kennesaw State University. She received a Ph.D. in Industrial Management at Clemson University, an MBA at Kennesaw State University, and a Bachelor's in Industrial Engineering from Georgia Tech. Her research interests include curriculum development and pedagogical improvements in the classroom, diversity in IT, and research methods in IS. She has published on these and other topics in *Journal of Computer Information Systems*, *Computers in Human Behavior*, and *Industrial Management & Data Systems*. She also co-edited *The Handbook of Information Systems Research*.

Tracy C. Guthrie received a Bachelor of Science in Information Systems from Kennesaw State University. He has over 20 years of information technology and management experience with two Fortune 500 companies. He is currently a Programmer Analyst/Project Lead at SunTrust bank.

Sherri Shade is Instructor of Information Systems at Kennesaw State University. She received a Master of Science in Information Systems and a Bachelor of Science in Information Systems, both from Kennesaw State University. Before joining KSU, Professor Shade spent many years in the software development industry. Her most recent experience was as Director of Education Services where she managed and implemented complex technical training, technical documentation and installation services for a proprietary POS software organization.