2007

Eliciting Software Requirements for Elementary Keyboarding Instruction

Mark A. Eirich

Eliciting Software Requirements for Elementary Keyboarding Instruction

Mark A. Eirich

November 19, 2007

Abstract

This Southern Scholars research project developed requirements for keyboarding instruction software to be used in the elementary classroom. The project was undertaken on the premise that the teachers in the Georgia-Cumberland Conference[1] are not satisfied with any currently existing keyboarding instruction software, and that software better suited to the classroom could be designed. Methods used to gather data included: a literature review on keyboarding instruction methods, a summary review of existing keyboarding software, interviews with teachers and school staff, and a survey of elementary teachers to evaluate their current situation and identify the software features they consider most important. These data were used to produce formal requirements documentation to aid in the development of new software or to evaluate existing software. Findings revealed that schools using Microsoft Windows have a number of quality keyboarding programs to choose from, while schools using Linux have extremely few. Recommendations were made for the development of a new product and for further research of instruction methodology.

Introduction

Keyboarding is one of the primary means of interacting with computers, and in an age when computers are an integral part of human life, it is an essential survival skill. Thus, the building of keyboarding technique and proficiency is an indispensable part of the school curriculum, and in recent years, has shifted from high school to the elementary school classroom.

In the classroom, it is possible for computer-assisted instruction to provide a number of advantages over traditional typing book instruction. Keyboarding software can provide immediate feedback to the student, automatically diagnose problem areas, offer games to motivate the student, adapt easily to an individual's prior ability and pace, and provide the teacher with accurate assessments.

The Georgia-Cumberland Conference[1] does not currently have a recommended keyboarding curriculum for its elementary schools, and teachers at each school have been left with the responsibility to choose materials and establish assessment standards. Most of the keyboarding programs they use are poorly suited to the classroom setting, and do not provide adequate centralized assessment and control. In addition, many of the schools are quite small and operate on very restrictive budgets. A number of them have switched from running Microsoft Windows to some form of Linux, and more are expected to transition in the near future. However, teachers have experienced difficulty finding quality keyboarding software that runs on Linux.

If designed carefully, software that meets the needs of the Conference should also meet the need of schools worldwide. It is reasonable to assume that public and private schools in developing countries and small schools everywhere share the need for low-cost, cross-platform

[1] Southern Adventist University lies within the Georgia-Cumberland Conference of Seventh-day Adventists, which oversees more than 50 elementary and high schools. The University's School of Computing and the Conference are actively seeking to establish a productive collaboration for the exchange of technology expertise and the development of valuable student experience.

keyboarding software that is specifically adapted to the elementary classroom. To allow the academic and open-source community the freedom to make ongoing improvements and to benefit the many schools that cannot afford proprietary software, the software should be licensed as free, open-source software.

Given the ubiquity of computers and hence the critical need for typing skills, the complexity of the multi-grade classroom environment found in many Conference elementary schools and the lack of high-quality, cross-platform keyboarding instruction software, it is clear that a very specific solution must be developed. However, as well-known software engineer Fred Brooks stated, "The hardest single part of building a software system is deciding precisely what to build. No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later" [1]. Thus, an in-depth requirements elicitation was undertaken. The objectives of this step were to determine the goals of keyboarding software as defined by teachers and education experts and to understand the way the software will actually be used in the classroom. It is possible that an existing program may be chosen after further review. However, though there are dozens of typing tutor programs available, most are incomplete solutions, and it is likely that a new product will be required.

Literature Review

The purpose of the literature review was to identify research studies that tested keyboarding instruction methodologies and made recommendations that could be implemented in software. However, recent research on the topic is virtually nonexistent.

Only a handful of research studies conducted since 2000 were found, and only one was relevant to keyboarding software requirements. The study involved 100 subjects and compared software with games and without, with and without hand covers. It concluded that students learned best while using hand covers together with software that uses games for

Introduction

Keyboarding is one of the primary means of interacting with computers, and in an age when computers are an integral part of human life, it is an essential survival skill. Thus, the building of keyboarding technique and proficiency is an indispensable part of the school curriculum, and in recent years, has shifted from high school to the elementary school classroom.

In the classroom, it is possible for computer-assisted instruction to provide a number of advantages over traditional typing book instruction. Keyboarding software can provide immediate feedback to the student, automatically diagnose problem areas, offer games to motivate the student, adapt easily to an individual's prior ability and pace, and provide the teacher with accurate assessments.

The Georgia-Cumberland Conference[1] does not currently have a recommended keyboarding curriculum for its elementary schools, and teachers at each school have been left with the responsibility to choose materials and establish assessment standards. Most of the keyboarding programs they use are poorly suited to the classroom setting, and do not provide adequate centralized assessment and control. In addition, many of the schools are quite small and operate on very restrictive budgets. A number of them have switched from running Microsoft Windows to some form of Linux, and more are expected to transition in the near future. However, teachers have experienced difficulty finding quality keyboarding software that runs on Linux.

If designed carefully, software that meets the needs of the Conference should also meet the need of schools worldwide. It is reasonable to assume that public and private schools in developing countries and small schools everywhere share the need for low-cost, cross-platform

[1] Southern Adventist University lies within the Georgia-Cumberland Conference of Seventh-day Adventists, which oversees more than 50 elementary and high schools. The University's School of Computing and the Conference are actively seeking to establish a productive collaboration for the exchange of technology expertise and the development of valuable student experience.

keyboarding software that is specifically adapted to the elementary classroom. To allow the academic and open-source community the freedom to make ongoing improvements and to benefit the many schools that cannot afford proprietary software, the software should be licensed as free, open-source software.

Given the ubiquity of computers and hence the critical need for typing skills, the complexity of the multi-grade classroom environment found in many Conference elementary schools and the lack of high-quality, cross-platform keyboarding instruction software, it is clear that a very specific solution must be developed. However, as well-known software engineer Fred Brooks stated, "The hardest single part of building a software system is deciding precisely what to build. No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later" [1]. Thus, an in-depth requirements elicitation was undertaken. The objectives of this step were to determine the goals of keyboarding software as defined by teachers and education experts and to understand the way the software will actually be used in the classroom. It is possible that an existing program may be chosen after further review. However, though there are dozens of typing tutor programs available, most are incomplete solutions, and it is likely that a new product will be required.

Literature Review

The purpose of the literature review was to identify research studies that tested keyboarding instruction methodologies and made recommendations that could be implemented in software. However, recent research on the topic is virtually nonexistent.

Only a handful of research studies conducted since 2000 were found, and only one was relevant to keyboarding software requirements. The study involved 100 subjects and compared software with games and without, with and without hand covers. It concluded that students learned best while using hand covers together with software that uses games for motivation [2]. However, games without hand covers or hand covers without games did not improve learning. More research would be needed to draw any definite conclusions.

The *Business Education Index* [3] was used to search for studies between 1987 and 2000. Directly applicable research was found to be very scarce. Somewhat relevant findings included:

- Self-directed, computer-assisted keyboarding instruction is at least as effective as teacher-directed instruction [4–9].

- Keyboarding skill and language arts skills are positively correlated [10–12].

Other topics included the establishment of speed and accuracy standards for postsecondary students [13], computer vs. typewriter-based instruction, and the opinions of teachers or professionals on the importance of keyboarding.

While there is little recent research on instructional methods, a large number of the articles and studies reviewed cited *Acquisition of Typewriting Skills* (1983) by Leonard J. West and *Teaching Keyboarding/Typewriting* (1984) by Gary N. McLean, a student of West [14, 15]. West's book is based on scores of studies conducted between 1920 and 1980. Both West and McLean list a large number of very practical guidelines for keyboarding instruction, and McLean lists 25 criteria for evaluating keyboarding software. These guidelines and criteria were incorporated into the requirements document for this project.

As noted by West, much instructional methodology in practice is based on lore and myth rather than research. For example, teachers often insist that students keep their eyes off the keyboard, even though research clearly indicates that looking at the keys is necessary for learning the keyboard [14,15]. Another example is the belief that rhythm is necessary for fast typing. West cites abundant research, as far back as 1923, to show that "the best typing is least metronomic," and concludes that "there is no place whatever in typewriting instruction for so-called rhythm drills." However, Mavis Beacon included metronome practice at least through 1999 [16].

Software included with a textbook published by McGraw-Hill has a highly sophisticated system to diagnose and remedy accuracy problems. It "analyzes and identifies 75 different types of possible misstrokes then prescribes specific, individualized drills to address each weakness" [17, 18]. It may be assumed that given the large investment required to develop such software, the publisher has a strong belief that such methods are useful. However, West cites a number of studies concluding that error analysis and corrective drills do not improve accuracy, and more recent research supporting such methods has not been found. Hence, developing such a feature should remain a very low priority until sufficiently supported by research. New software should not necessarily imitate existing software, regardless of its popularity.

Software Review

The goals of the software review were to identify common features of keyboarding software to aid in the development of requirements, to discover free software that could be extended or incorporated into a new product, and to compile a list of programs to facilitate the selection process in the event that a new product is unnecessary or not feasible.

Software for personal keyboarding practice has existed about as long as the personal computer. Some examples are Microsoft Typing Tutor, which was introduced in December 1979 for the TRS-80 [19], and Mavis Beacon, which was introduced in 1987 [20] and ran on the Commodore 64 [21] and the Apple II [22]. Today, there are dozens of programs available. Utah State Office of Education has rated over 125 different keyboarding programs [23].

In reaction to the many abuses of the proprietary software industry, many of the best software developers now write free software (free as in *libre*, not *gratis*), and the number of quality programs available as free software has rapidly increased. Free software is of particular interest because anyone is free to download the source code and make improvements or modify it in any way [24]. Thus, any keyboarding program released as free software, even if it is not suited to the elementary classroom, may be extended to meet the requirements.

SourceForge.net and FreshMeat.net list tens of thousands of open source software and free software projects[2]. A search of the two sites returned 39 keyboarding projects. Of these, at least two were not free, three projects had not yet released any software, and four did not appear to offer an English version. Of the remaining 30, 11 were console-based (providing a fairly spartan user interface), and 19 had graphical user interfaces. One of the more sophisticated programs, KTouch [26], as well as a number of others, are not available in Windows. Four of the 30 were games only. The games and console-based programs were not reviewed.

In general, the features offered by the free software reviewed were very limited. All offered very little practice material; after a few months a teacher would need to either enter new lessons or switch to another program. One exception to this is N-Type, which downloads BBC news articles on the fly [27]. However, N-Type does not offer a structured lesson format, nor is the source code available for modification. Even worse than the lack of practice material was the lack of instruction. Almost none of the programs provided explanations or demonstrations of any kind. The exception is HyperType, which displays a single sentence at the beginning of each new key lesson to explain which finger should be used [28].

None of the free software reviewed seemed to be designed for kids. They offered no variety to make learning fun and interesting, but presented only a single screen layout and a single type of practice. Their graphic user interfaces were very plain and unattractive. The few that offered games did not allow any teacher control of how frequently the student is

[2] The terms "open source software" and "free software" represent two distinct philosophical approaches to software licensing [25]. The distinction is not prerequisite to an understanding of this paper but merely an acknowledgment of the diversity of licensing.

permitted to play them. Many display a keyboard on the screen, but the keyboard cannot be hidden by a teacher or lesson designer for certain lessons. Only HyperType displays a "shadow" of hands typing on the keyboard.

It is assumed that in the school setting, a teacher needs some basic student progress information, but this functionality was largely lacking. TypeFaster [29] offers the only centralized teacher interface, allowing creation of different lesson sequences for different grades, but does not display any progress history. KTouch displays the best student progress graphs, but does not make it accessible to a teacher. To access the information, a teacher would need to go to each computer in the lab and log in to the operating system repeatedly for each student.

In summary, the free software reviewed offered a few interesting features, but none seemed suited for the elementary classroom. However, the source code of some should be examined and the basic logic borrowed when a new program is developed.

An in-depth review of existing proprietary keyboarding software was not possible within the time constraints of this project. There are indexes and reviews available from a number of online sources, including Utah State Office of Education [23], SuperKids Educational Software Review [30], KnowPlay Educational Technologies [31], and TopTenREVIEWS [32, 33]. A small sample was examined to identify common features.

Unlike most free software, most of the proprietary programs were available only for the Microsoft Windows operating system. Of the sample reviewed, TypingMaster was the only commercial program available for Linux [34]. A number of programs were Macintosh-compatible.

While many typing programs target personal use by adults, many are specifically designed for children and for classroom use. At least eight programs reviewed were intended for school use. Some of the management features they offered were:

- Management of students by class

- Graphing and charting of student progress data
- Importing and exporting of student names or progress data
- Printing or exporting detailed or summary reports for an individual or for the class
- Assignment of grades
- Customization of lesson content
- Control of the frequency of game playing
- Adjustment of accuracy and speed goals

The programs for younger students were quite graphics-intensive, with a look, feel, and vocabulary intended to make learning fun. Typical features included:

- Practice material taken from children's stories, nursery rhymes, and fairy tales
- A variety of games
- Attractive, graphics-filled screens
- Animated demonstrations of typing technique
- Realistic on-screen keyboard with 3-D animated hands
- Lessons presented as missions, challenges, or adventures

In addition to keyboarding instruction software installable in the traditional way, a respectable variety of online programs were discovered. The nineteen web sites reviewed varied widely in content, cost, and quality. Four of the sites contained mostly typing games and little instructional content. There are at least twenty different free typing games available online, with varying levels of quality and difficulty. Four of the nineteen sites required a paid subscription.

Quality varied dramatically. One site that purported to have lessons for all ages was crowded with banner advertisements for online dating web sites. On the other extreme, BBC's free (but not open-source) Dance Mat Typing teaches kids through an interface as interesting and fun as any off-line kid's program [35]. It covers the entire keyboard in a lively, varied, and interactive set of twelve lessons. While not free, Custom Solutions' Custom Typing Training also offers schools a program that is well-suited to kids [36]. Unlike Dance Mat, Custom Typing provides a class management interface for the teacher, with the usual progress graphs, reports, and lesson customization features.

The online typing programs reviewed clearly demonstrated that web technologies such as Javascript and Adobe Flash are fully capable of delivering high-quality keyboarding lessons. Some of the benefits of using online programs are:

- The program can be used on *any* computer equipped with a web browser and Flash plug-in; no additional software needs to be installed
- Students with internet access at home can practice at home
- Software upgrades do not need to be distributed
- Lesson designers can distribute new materials instantaneously

In summary, a bewildering assortment of keyboarding software is available. Second, free software has not yet reached the level of development as has proprietary software, but a variety of functionality exists that could built upon or incorporated into new software. Finally, though the selection is still small, online programs are a viable and attractive option.

Methods

Several research methods were used to develop the requirements for a program that would be actually useful in the elementary classroom. Besides the literature and software review, the following methods were used to gather data: **several** interviews with teachers and computer support, an *ad hoc* keyboarding curriculum committee organized by the Conference, and a survey conducted under the direction of that committee.

The interviews were conducted to establish a context for the rest of the research. The six participants were: a 5th-8th grade teacher from a school of 23 students (school A); a 6th grade teacher, a school secretary and parent, and a principal from a school of 98 students (school B); an education technology director and teacher from a school of 435 students (school C); and a computer specialist who supports a large number of the Conference's elementary schools.

The teachers, secretary and principal were asked questions to ascertain the following: the level of emphasis placed on keyboarding, what grades receive keyboarding instruction, the methodology employed, the level of success experienced by teachers and students, and any specific difficulties encountered. The computer specialist was questioned regarding some the specifics of what technology was available in school computer labs, frustrations observed or expressed by teachers about keyboarding software, and specific shortcomings of software currently in use.

Not long after the commencement of this project, the researcher was appointed by the Georgia-Cumberland Conference Curriculum, Instruction, and Assessment Committee to serve on an *ad hoc* Keyboarding Curriculum Committee. The purpose of the committee is to select or create a keyboarding curriculum for the schools operated by the Conference.

Other members of the committee included the Conference's three Regional Directors, an Assistant Professor of the School of Computing at Southern Adventist University, and six principals and teachers. One of the teachers is also a member of the North American Division K–12 Technology and Distance Education Committee and is involved in the Southern Union Educational Technology Association. Each teacher had experience teaching keyboarding.

At its first meeting, the committee voted to survey the principals and teachers of the Conference schools, and to assign the task of developing the instrument and analyzing the results to the author of this paper. The basic purpose of the survey was to identify what programs are being used and to discover what program components teachers find most useful or desirable. Electronic mail (email) was chosen as the medium because teachers are accustomed to it as the primary means of communication with the Conference administration.

After review by two professors of the School of Computing, the survey was sent to the members of the committee for further review. The chair of the committee approved the survey and sent it via email to each of the Conference's 156 elementary teachers. The note accompanying the survey requested that the teachers respond by replying to the email within a week. The chair of the committee would subsequently forward the responses to the researcher for tabulation and analysis.

The survey instrument consisted of twenty questions presented in an easy-to-read format. An introductory note explained briefly the goals of the committee, the purpose of the survey, and that teachers not involved in keyboarding instruction need not respond. These types of items were included:

- A question asking what grades the teacher teaches, to set a context for the other responses.

- Three questions about the amount of time students spend in keyboarding, and any reasons why they are not able to spend as much time as the teacher desires, and the importance of keyboarding relative to other subjects. The purpose was to determine appropriate lesson length and explore the possibility that shortcomings result from the low priority given to keyboarding rather than deficiencies in keyboarding software.

- Two questions about the style of keyboarding instruction—whether students progressed at their own rate or with the class, and whether students practiced concurrently or at various times of the day. Responses could have a large impact on curriculum structure.

- A multiple-choice question about the operating system used on the school's computers, to determine what technologies should be used in software designed.

- One question that asked what printed material, if any, the teacher uses, primarily to ascertain whether teachers are combining textbook material with software-based instruction.

- Six questions regarding keyboarding software currently used—its name, desire to replace it, the level of satisfaction, and specific features found useful and shortcomings found frustrating. The results could be used to develop requirements or as reviews of existing software.

- Three questions about how progress is monitored and problem areas discovered, to determine what software should accommodate.

- A list of 15 keyboarding software features. Respondents were asked to imagine that they were designing a new keyboarding program and mark the features as essential, useful but not critical, or unnecessary. Responses will guide in the prioritization of software requirements.

Taken together, these research methods form the basis for software development. They support an understanding of the characteristics and dynamics of the classroom environment, the nature of the problem from the eyes of the teacher, and the way they will interact with the software.

Results

Despite the informal nature of the interviews conducted, they provide important insight into several practical aspects of the problem.

Even in the small sample interviewed, the amount of time spent teaching keyboarding varied substantially. In one school, keyboarding instruction takes place in a time period as short as 10 minutes, while at another, keyboarding class may last up to one hour. *Thus, lessons must be either quite short or be interruptible. For schools with long periods, the amount of variety in material will be more critical, as attention must be maintained for up to 40 minutes of software-based instruction.*

All teachers interviewed pointed out that students do a significant amount of typing outside of keyboarding class. For example, interviewees from school B said that computers were integrated into the rest of the curriculum, and that students often used word processing and presentation software to complete assignments. The interviewee from school C encourages parents to purchase typing software so that their kids can practice at home. *Developers should consider creating software to monitor typing in other programs and design software that can be accessed both from home and school.*

Two schools do not have an in-house computer expert and had recently experienced major disruptions in the availability of their computer labs. The result was that for a few months, no keyboarding was taught at all. In the words of one teacher, "I'm not computer savvy enough to know what to do when something goes wrong." When computers "die," students may have to begin again at lesson one. The same teacher said that she had not been using the keyboarding software because there was not a shortcut for it on the desktop. *Software should be extremely robust and reliable, and all aspects of installation and use should be simple enough for a computer illiterate. Again, consider online software that can access student status from any location, and that requires no installation.*

None of the teachers expressed any major frustrations with the software they currently use. Schools A and B use Mavis Beacon Teaches Typing for the upper grades. One teacher noted that Mavis Beacon does not limit game playing, and unless carefully watched, some students will avoid completing lessons. Schools A and C use separate programs for third and fourth grades which are better suited to younger students. School B does not currently teach keyboarding below fifth grade. *Unless it is remarkably good or inexpensive, teachers may not have any motivation to switch to a new program. Software must keep students on track and be adaptable to the age of the student.*

The teachers at schools A and B pointed out that there exist very large gaps of proficiency among their students. One teacher stated that some had learned how to type previously, and now attain speeds of up to 60 wpm, while others struggle at 15 wpm. School C, which is larger and has alternate activities available, allows students to take a mastery test. Those who pass at 35 wpm do not have to continue in keyboarding. *Software must have appropriate lesson material for students of all proficiency levels.*

The teacher at school C heavily emphasized teacher involvement. The teacher should carefully observe students so they do not form bad habits. Keyboarding software cannot detect habits such as using the wrong key. Above all, students need abundant encouragement and praise. *Software should display frequent reminders of technique. Also, the software should set attainable goals and congratulate the student upon their completion.*

According to the support specialist, five Conference schools operate Linux on the desktop, and almost every school has a Linux server. He pointed out that Mavis Beacon is used in most of the schools, despite the fact that it lacks a number of critical class management features. Most of the features he mentioned are included in a number of Windows-based programs reviewed, but not in any Linux-based programs. However, he suggested one feature not found in any program reviewed: network-based activities for full-class participation.

At the time of this writing, the keyboarding committee had met only once, and little new data was collected. Of some interest was a draft copy of "Keyboard and Computer Literacy Suggestions for Consideration" under development by the Southern Union[3]. It

[3] The Southern Union Conference is an administrative division of the Seventh-day Adventist Church including Alabama, Florida, Georgia, Kentucky, Mississippi, the Carolinas, and Tennessee.

outlines curriculum requirements and may be useful for developing lesson sets.

Perhaps the most critical fact revealed at the committee was that while the Southern Union has set a formal standard to be reached by students completing eighth grade (30 wpm and <1 epm), and required that wpm be reported on student progress reports, no further guidelines or recommendations of any sort are available to teachers. It should thus be well noted that any solution focused solely on software will likely have little impact.

Unfortunately, by the time of this writing only 12 survey responses had been received. Further changes to the requirements may be made as more responses are tabulated. Though the sample is insufficient to represent the population, some observations were made.

Four of the respondents teach in schools with 80-100 students, while the remaining eight respondents teach in schools with an average of less than 12 students. Half report barriers to teaching as a result of an insufficient number of computers or other computer problems. Two thirds report that students practice keyboarding separately at various times during the day instead of concurrently as a class.

Half use Mavis Beacon, while none use printed material. Surprisingly, a teacher from one of the larger schools uses `http://freetypinggame.net/` and is very satisfied. Contrary to the premise of this paper, all but one responded indicating a high level of satisfaction with the software currently in use.

The most popular program features were:

- Automatically detect a student's problem areas and assign drills accordingly.
- Provide the teacher with convenient access to meaningful progress information for all students.
- Show a keyboard with hands on the screen.

The least popular were:

- Frequently remind students of proper posture and technique using pictures or video.
- Track typing in other programs (such as Microsoft Word) and provide additional training for difficult keys and words.
- Allow the teacher to create custom lessons and drills.

Discussion

Each research component provided information which aided in the formation of the requirements. Initial requirements were written based on the features of software reviewed and observations from interviews. These requirements were then refined and prioritized based on findings from the literature review and teacher survey.

Contrary to an assumption of this project, teachers seem fairly satisfied with the software they are currently using. Also, for schools using Microsoft Windows, there are many high-quality keyboarding programs available. However, quality elementary keyboarding software for Linux is virtually nonexistent. Schools using Linux may choose from the very few decent online programs, but unless teachers are satisfied with these, the Conference should commission the development of a new program.

Regardless of whether new software is developed, the Conference should publish a list of recommended software. Teachers, especially those in smaller schools, do not have time for such research. If a new program is not created, the Conference should consider purchasing volume licenses of several programs to make them more affordable to small schools.

A feasibility analysis should precede the decision to create new software. The development of high-quality software is labor-intensive; the credits page for Type to Learn 3 lists over 35 developers, including graphic designers, artists, writers, editors, sound engineers, and software engineers [37]. Southern Adventist University students may provide the least expensive labor available for the project. The team would ideally be comprised of students studying elementary education, art, graphic design, animation, English, and computing. To attract participants, faculty should give students working on the project the opportunity to receive credit in required courses, or as an internship elective. In addition, the Conference could offer scholarships to team members who accomplish project objectives. Other developers and advisors should be recruited by contacting maintainers of open-source keyboarding projects and by launching an internet publicity campaign.

To be successful, the lessons must be designed to integrate seamlessly into the Conference's keyboarding curriculum. To encourage adoption, the new software should be actively marketed to teachers and endorsed by superintendents. Also, a qualified instructor should conduct training workshops at teachers' and principals' meetings.

Some questions that could be asked in further research include:

- Are corrective drills based on error analysis effective in skill-building?
- What specific types of practice (i.e. progressive, paced, sprints) are most effective?
- What specific types of copy material (i.e. stories or random words, low or high syllabic intensity) are most effective?

Bibliography

[1] F. P. Brooks, "No silver bullet: Essence and accidents of software engineering," *IEEE Computer*, vol. 20, no. 4, pp. 10–19, 1987.

[2] S. D. Reagan, "Increasing touch-keyboarding skills in the middle school student: 'Key-Words' vs. 'Type to Learn,' hand covers vs. no hand covers," Master's thesis, Johnson Bible College, 2000, ERIC (full text) ED443386.

[3] C. L. Noll and P. R. Graves, Eds., *Business Education Index.* Delta Pi Epsilon Society, Little Rock, AR, editions 1987-2000 available on ERIC except 1991.

[4] I. Russin, "A comparison of the effect of teacher-directed instruction (and textbook use) and interactive computer software instruction on the development of touch-keyboarding skills in two sixth-grade classes," Master's thesis, Kean College, NJ, 1995, ERIC (full text) ED381132.

[5] K. G. Buhs, "A comparison of the effectiveness of teacher-directed instruction and independent study of computer keyboarding skills for students in grades three through five," Ph.D. dissertation, Univ. of Miami, 1996, pub. no. AAT 9628753, ProQuest (abstract) 742552311.

[6] E. M. Crowell, "A comparative analysis of teacher-directed and computer-assisted instruction in terms of students' keyboarding achievement, attitudes, and cooperative

learning," Ph.D. dissertation, Michigan State Univ., 1992, pub. no. AAT 9314659, ProQuest (abstract) 746278561.

[7] J. E. Toppe, "A comparison of teacher-directed instruction and review vs. computer-assisted instruction and review in the development and maintenance of keyboarding skills at the fifth-grade level," Ph.D. dissertation, Indiana Univ., 1990, pub. no. AAT 9109765, ProQuest (abstract) 745123481.

[8] R. J. Barta, "A comparison of teacher-directed computer-based instruction and self-directed computer-based instruction in teaching beginning keyboarding to college students," Ph.D. dissertation, Univ. of Minnesota, 1987, pub. no. AAT 8723808, ProQuest (abstract) 753036381.

[9] B. A. Hauger, "A comparison of computer-assisted instruction and teacher-directed instruction in teaching microcomputer keyboarding to sixth-graders," Ph.D. dissertation, Univ. of North Dakota, 1986, pub. no. AAT 8713240, ProQuest (abstract) 753006441.

[10] M. C. Sowers, "The effect of keyboarding instruction on language arts and keyboarding skills of third-grade students in the Wichita Public Schools," Ph.D. dissertation, Oklahoma State Univ., 1989, pub. no. AAT 9019519, ProQuest (full text) 746388861.

[11] K. A. Wetzel, "The effect of using the computer in a process writing program on the writing quality of third, fourth, and fifth grade pupils (word-processing, language, elementary, composition, keyboarding)," Ph.D. dissertation, Univ. of Oregon, 1985, pub. no. AAT 8605868, ProQuest (abstract) 752600491.

[12] J. E. Shalaby, "A study of the relationship of spelling ability to the acquisition of keyboarding speed, accuracy, and proofreading skills of selected seventh- and eighth-grade students," Ph.D. dissertation, Rutgers New Brunswick, 1994, pub. no. AAT 9514126, ProQuest (abstract) 741434091.

[13] V. Arnold, R. L. Joyner, B. J. Schmidt, and C. D. White, "Establishing electronic keyboarding speed and accuracy timed-writing standards for postsecondary students," *Business Education Forum*, vol. 51(3), pp. 33–38, 1997.

[14] L. J. West, *Acquisition of Typewriting Skills.* Macmillan Publishing Co., Inc., Indianapolis, IN, USA, 1983.

[15] G. N. McLean, *Teaching Keyboarding/Typewriting. Delta Pi Epsilon Rapid Reader No. 5.* Delta Pi Epsilon, St. Peter, MN, 1984, ERIC (full text) ED250500.

[16] J. D. Biersdorfer, "Library/typing tutors; it's like typing class, but you set the rhythm," *New York Times*, feb 1999, http://query.nytimes.com/gst/fullpage.html?res=9D04E0D81E3AF93BA25751C0A96F958260.

[17] S. Ober, J. E. Johnson, and A. Zimmerly, *Gregg College Keyboarding and Document Processing*, 10th ed. McGraw-Hill Irwin, 2006.

[18] A. Zimmerly. What's new in GDP 10. PowerPoint presentation. [Online]. Available: http://gdpkeyboarding.com/PowerPointZimmerly/What's_New_GDP10.ppt

[19] K. Polsson. Chronology of personal computers. [Online]. Available: http://www.islandnet.com/~kpolsson/comphist/comp1979.htm

[20] J. D. Biersdorfer, "Next they'll say Betty Crocker isn't real, either," *New York Times*, December 1998, http://query.nytimes.com/gst/fullpage.html?res=9900E7DC133FF932A05751C1A96E958260.

[21] K. Polsson. Chronology of the Commodore 64 computer. [Online]. Available: http://www.islandnet.com/~kpolsson/c64hist/c641988.htm

[22] S. Weyhrich. Apple II history. [Online]. Available: http://apple2history.org/history/appy/aha88.html

[23] K-12 keyboarding. Utah State Office of Education. [Online]. Available: http://www.schools.utah.gov/ate/keyboarding/Software/ALLsoft.htm

[24] (2007, jun) GNU general public license. Free Software Foundation, Inc. [Online]. Available: http://www.gnu.org/copyleft/gpl.html

[25] Wikipedia contributors. (2007) Alternative terms for free software — Wikipedia, the free encyclopedia. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Alternative_terms_for_free_software&oldid=170083516

[26] A. Nicolai and H. Frøiland. KTouch — free touch typing tutor. [Online]. Available: http://ktouch.sourceforge.net/

[27] N-Type. Riff-Raph Development Team. [Online]. Available: http://n-type.sourceforge.net/

[28] J. Nisly. HyperType. [Online]. Available: http://www.hypertype.org/

[29] D. le Roux. TypeFaster typing tutor. [Online]. Available: http://www.typefastertypingtutor.com/

[30] Typing software — the big picture. SuperKids Software Review. [Online]. Available: http://www.superkids.com/aweb/pages/reviews/typing/

[31] Typing software. KnowPlay Educational Technologies. [Online]. Available: http://www.educational-software-directory.net/training/typing.html

[32] Typing software review 2008. TopTenREVIEWS, Inc. [Online]. Available: http://typing-software-review.toptenreviews.com/

[33] Typing for kids software review 2008. TopTenREVIEWS, Inc. [Online]. Available: http://typing-for-kids-software-review.toptenreviews.com/

[34] TypingMaster Typing Tutor. TypingMaster, Inc. [Online]. Available: http://www.typingmaster.com/

[35] Dance Mat Typing. BBC. [Online]. Available: http://www.bbc.co.uk/schools/typing/

[36] Custom typing training. Custom Solutions. [Online]. Available: http://www.customtyping.com/

[37] E. Rappoport, Ed., *Type to Learn 3 Teacher's Guide.* Sunburst Technology Co., 2002, http://store.sunburst.com/images/TeacherGuide/TTL3_Guide.pdf.

~~Mark's cop~~,

cholars Honors Program
ject Proposal Information Sheet

Mark Ginich

Date Sept. 12, 2007

SOUTHERN
ADVENTIST UNIVERSITY

Computer Science

Southern Scholars
southernscholars.southern.edu
wmclarty@southern.edu

Aant scholarly project, involving research, writing, or special pence, appropriate to the major in question, is ordinarily completed thr year. The project is expected to be of sufficiently hiality to warrant a grade of "A" and to justify public presentation.

Unhe guidance of a faculty advisor, the Senior Project should be an original work, should use primary sous when applicable, should have a table of contents and works cited page, should give convincing eviace to support a strong thesis, and should use the methods and writing style appropriate to the discline.

<u>The completed project, to be turned in in duplicate, must be approved by the Honors Committee in consultation with the student's supervising professor four weeks prior to the last day of class for the semester the project is turned in.</u> Please include the advisor's name on the title page. The 2-3 hours of redit for this project is usually done as directed study or in a research class.

IOTE-Senior Project Proposal Due Date: The senior project proposal is due in the Honors Program irector's office two weeks after the beginning of the semester the project will be completed. The proposal ould be a detailed description of the Honors Project's purpose and proposed methodology.

eeping in mind the above senior project description, please describe in as much detail as u can the project you will undertake. Attach a separate sheet of paper.

nature of faculty advisor W Munger

ected date of completion Nov. ~~19~~ 16, 2007

E: An advisor's final project approval does not guarantee that the Honors Faculty Committee will natically approve the project. The Honors Faculty Committee has the final vote.

Approval to be signed by faculty advisor when the project is completed:

Throject has been completed as planned (date) Nov 19, 2007

This an "A" project Yes

This bject is worth 2-3 hours of credit 3

Advisr's Final Signature W Munger Date: Nov 19, 2007

Chair, Honors Committee ______________ Date Approved: __________

Dear Advisor,

(1) Please write your **<u>final</u>** evaluation on the project on the reverse side of this page. Comment on the characteristics that make this "A" quality work.

(2) Please include a paragraph explaining your specific academic credentials for advising this Senior Project.

See Attached

This project is a report of an on going software development process. It is very typical of computing science research to have a partial or interim report. This report describes the process used for the development of the requirements for a program to assist in the teaching of keyboarding at the elementary school level in the Georgia Cumberland Conference. Such a process is unique due to elementary school setting and the lack of a low-cost cross-platform program to aid in teaching keyboarding. The uniqueness of the setting for the keyboarding program requires a combination of several methods and tools to adequately develop a software requirements document.

Most conference papers will allow only 5 to 8 pages. We have chosen to use the IEEE format.

This paper represents 3 credit hours worth of work. Both the work and the paper are excellent and are of "A" quality.

My qualifications for advising this Senior Project:

1. I have published software engineering conference papers.
2. I have taught Requirements at both the graduate and undergraduate levels.
3. I have taught our Senior Seminar Class, a writing course.
4. I have developed educational software.

W Munger
Nov. 19, 2007