University of Alberta

Computer-Assisted Instruction in the Primary Grades:

Which Authorized Software Tools Help to Teach Reading and Writing?

by

Meridith Ann Lovell ©

A thesis submitted to the Faculty of Graduate Studies and Research

in partial fulfillment of the requirements for the degree of

Master of Education

Department of Elementary Education

Edmonton, Alberta
Spring, 2008

Your file Votre référence
ISBN: 978-0-494-45754-2
Our file Notre référence
ISBN: 978-0-494-45754-2

Abstract

Computer-assisted instruction, or the use of computers to enhance aspects of traditional instruction, is one facet of the integration of technology into classroom instruction. A list of authorized software for teaching reading and writing in Canada was compiled and 13 software titles (28%) were assessed using an adapted form of Bishop and Santoro's rubric (2005) for evaluating early reading software. Programs were assessed for Interface Design, Content, Instructional Design, and whether the manufacturers' educational claims were supported by software functions. Four software groups emerged: *Reading Programs*, *Writing Programs*, *Programs for Students with Special Needs*, and *Other Programs*. Many of the 13 analyzed programs were non-instructional tools that merely replaced reading and writing functions traditionally performed manually. Several of the analyzed programs were outdated. These findings call into question how provincial and territorial mandates to meaningfully integrate technology into instruction, especially reading and writing instruction, are being met across Canada.

Acknowledgements

I am reminded, now that all of the research, writing, and re-writing has finally been completed, of all of the support and guidance I have received while writing this thesis. Although it says "by Meridith Lovell" on the very first page, without the support of my advisor and examination committee, my colleagues, and my family throughout the writing process, this work would never have been successfully completed.

First, to my supervisor, Dr. Linda Phillips, who patiently read through everything I wrote with her meticulous attention to detail, offered suggestions and guidance to improve my ideas, and gave me a much needed kick whenever I strayed too far from where I needed to be: thank you. Those two words seem inadequate for conveying the depth of my gratitude for all of your assistance, yet you have taught me to be concise and say exactly what I mean, and so, thank you.

To Dr. Carol Leroy, who served as Chair of my examination committee: I have always thought of you as another mentor throughout my degree. I am very grateful for the way in which you always challenge me to look at everything from new perspectives.

To Dr. Patricia Boechler, who served as my external reviewer: we have only recently met, but I wanted to thank you for the points you raised during my examination. They will certainly give me new perspectives to consider in the future.

Thanks are also due to: the contacts at the publishing houses and Ministries of Education, who answered all of my questions; my colleagues, who offered emotional support; Christina Hinger, who helped perform the inter-rater reliability testing; and Josie Nebo, who helped track down all of the software for me.

As always, final thanks go to my family, for always believing that I can do whatever I set my mind to and pushing me to actually go and do it.

Table Of Contents

List of Tables

CHAPTER 1: INTRODUCTION

In many classrooms and most schools across Canada, the presence of computers is an accepted part of the school experience, and "… most educators agree that computer access and literacy have become vital and necessary for young learners in the 21st century" (Judge, Puckett & Bell, 2006, p. 52). The purpose of this study is to examine the Ministries of Education authorized software for use in the primary grades to teach reading and writing.

Background and Theoretical Framework

In the past, computer classes in the primary grades were taught separately from other curricular areas and often students would not see or use a computer between these classes. Recently, the availability and use of computers within schools and within society in general has increased significantly, as has the need for students to be more technically literate. In Canada, many programs of study, such as the Alberta Program of Studies, call for the integrated use of technology within the classroom, and many of the technology outcomes deal specifically with computer use (Alberta Learning, 2000), requiring that students have adequate access to computers and use them meaningfully for learning throughout the curriculum. When computers are meaningfully integrated into the learning experiences of primary students, the potential exists for computers to become powerful learning tools. Computers can promote "… basic and higher level literacy skills needed for full participation in contemporary society" (Littleton, Wood & Chera, 2006, p. 382). Among other benefits, computer software programs permit students to work at individual paces, to receive feedback quickly, and to enjoy multimedia learning experiences.

Increased time on task, increased motivation and lower costs are also benefits associated with computer usage (Norris, Sullivan, Poirot, & Soloway, 2003).

A multitude of software programs are approved for use in Canadian classrooms and claim to assist children in the areas of reading and writing development, yet not much is known about how such programs promote literacy development through reading and writing and their selection and use raise several questions. What software is available to promote reading and writing development in the early grades? Who are these software programs designed by and who are they designed for? What claims are made about how these software programs promote reading or writing development? Do these software programs actually do what they claim? How do they work? How do the objectives and outcomes of these software programs align with the philosophy statements and learner outcomes found in the Language Arts Programs of Study and the Technology Programs of Study? Are these software programs available to a wider audience than schools, or are they designed specifically for classroom or school use? Do these software programs assist in classroom instruction, claim to replace classroom instruction, or do they supplement traditional Language Arts programs? Are they appropriate for the developmental levels of the children for whom they are designed? Are the programs proprietary, that is are they linked only to one publisher's resources, or do they "stand alone"? What do they cost? What benefits do students gain and do they actually learn from these software programs? These questions, and many more, need to be considered when evaluating the usefulness of computer software supposedly designed to promote reading and writing development in the early grades.

Whether purchasing new software for classroom or school-wide use, or evaluating existing software for effectiveness, final decisions about what to purchase or what to use are constrained by monetary concerns. New programs are expensive. Fees for purchasing new software and maintaining existing site licenses can consume large portions of school technology budgets. Newer programs may be overlooked due to cost while older programs may remain in the school for the same reason, even if they are neither effective nor aligned with the outcomes and philosophy of the Program of Studies.

Teacher perception also influences how computers are used in classrooms. If teachers are comfortable with computers and feel prepared and supported in teaching with technology, they will typically feel more comfortable using technology in their classrooms (Wozney, Venkatesh & Abrami, 2006). Is computer time treated as free time, or is it used to meaningfully support the Language Arts program? Do students in the early grades use computers meaningfully or only for drill and practice activities (Franklin, 2007)? Will students, teachers, administrators, and parents see new software as simply a game or will they see the potential instructional benefits for reading and writing development? Critically evaluating software that is designed to assist in reading and writing development will serve to clear up some of the confusion about what is available, what it does, and whether it is effective for teaching reading and writing skills so that informed decisions can be made about which programs to use in schools.

Purpose and Significance of the Research

The purpose of this research is to examine and evaluate the authorized resources for computer-assisted instruction in reading and writing in the early grades by highlighting the benefits and drawbacks of these various computer software programs that claim to teach, or assist in teaching reading and writing skills, and then to examine how they may be used to complement reading and writing instruction within schools. The goal is to establish a benchmark of the various types of programs available and to investigate how they are intended to work. The proliferation of teaching software and confusion over what is available and what is effective for classroom or school use may mean that the most effective programs are not being used, and that the programs which are being used are not helping in the ways they claim.

During the primary grades, effective reading and writing instruction is essential for students to learn the skills to support further academic growth. Software programs have been developed and evaluated to help at-risk students develop reading and writing skills (for example, see Bishop & Santoro, 2006; Fasting & Lyster, 2005), yet these authors did not evaluate software designed for regular classroom use. Each of the provincial Ministries of Education have a limited list of authorized software for assisting in reading and writing instruction, thus a systematic review of these programs will allow further discussion of their merits for use in Canadian classrooms and will identify patterns present in software selected as authorized resources across Canada.

Research Questions

In order to investigate the effectiveness of available software, it was necessary to research the various kinds of programs available (proprietary software, drill and practice software, games, etc), how widely these programs are available and their claims about what they can accomplish. Investigating what the professional literature says about the place of computer-assisted instruction in reading and writing in the early grades and the issues and questions raised about computer use adds depth to the multitude of questions I have raised in previous sections so that they may be grouped into broader categories for research purposes:

1) What types of authorized software are available?
2) What do the manufacturers claim their products do and do the products support the claims?
3) Are reading and writing prerequisite skills for using the software programs?
4) Are these software programs appropriate and useful to supplement classroom reading and writing instruction?
5) What issues or trends influence software selection and use?

The answers to these questions should help to clarify some of the confusion surrounding what is available and how the available programs might support reading and writing instruction in primary classrooms.

Definition of Terms

Computer-Assisted Instruction is the use of computer software to assist in the teaching of skills or content information. The degree of software use varies from complete instruction by the software to having the software support classroom instruction.

Desktop Publishing refers to software programs whose primary function is to allow users to incorporate text and pictures into various layouts. Most desktop publishers incorporate limited word processing features.

Pixel refers to the dots that make up a digital image. High-resolution graphics have smaller pixels, contain more pixels in each square inch, and approximate the way the eye views objects in real life. Low-resolution graphics have fewer pixels per square inch, appear less defined and life-like, and are referred to as *pixelated.*

Proprietary Software is software that is linked specifically to a publisher's resources. These software programs are not designed to be used independently of the linked resource, and therefore do not "stand alone".

Text-to-Speech software uses computer-generated speech to read what is printed on the computer screen. Some text-to-speech programs highlight each sound or word as it is read.

Word Processor refers to software programs whose primary function is to allow users to enter, modify, and format text and so that it may be printed in a format resembling what is displayed on the computer screen. Most word processors allow limited picture and text layout options.

Overview of Thesis Organization

Chapter one provided an introduction. Four chapters follow which explain my research and conclusions. Chapter two examines the relevant research literature on the use of computers in primary classrooms and the use of software to teach or assist in teaching reading and writing. The methodology used is described in Chapter three and the data are presented and analyzed in the fourth chapter. The fifth chapter provides conclusions and recommendations.

CHAPTER 2: REVIEW OF THE RELEVANT LITERATURE

In the last decade, computer use in Canadian schools and within society has increased dramatically. In Canada 67.9% of adults use the Internet (Statistics Canada, 2006b) with 80.9% of adults with minor children reporting that they use the Internet (Statistics Canada, 2006a), and this is one indicator of the prevalence of computer use in society. Computers have the potential to be very helpful and powerful tools for teaching or assisting in the teaching of many content areas. Access to computers and the ability to use computers successfully "...will greatly help young people become productive and engaged citizens in the 21st century" (Judge, Puckett & Bell, 2006, p. 59). In fitting with this view, many programs of study, such as in Alberta (Alberta Learning, 2000), call for the meaningful integration of technology and computers into other curricular areas. In the example of Alberta, "The ICT curriculum is not intended to stand alone, but rather to be infused within core courses and programs" (Alberta Learning, 2000, p. 1). This statement is echoed in the programs of study for other provincial and territorial jurisdictions as well. For instance, in Prince Edward Island, (Prince Edward Island Department of Education, 2006, p. 15) the rationale of the Communication and Information Technology (CIT) Integration curriculum is

> ... to focus on how CIT can be used from grades 1-6 and across all areas of the curriculum as part of a more global strategy that will contribute to the development of technologically competent and literate individuals graduating from our school system. As technology is best learned within the context of applications, activities, projects, and problems that replicate real-life situations, the CIT program of studies is structured as a 'curriculum within a curriculum',

using the core subjects of English Language Arts, Math, Science and Social Studies as a base.

The Yukon Territory, which adapted its Program of Studies from Alberta's, includes this statement, "This curriculum is not intended to stand alone. It has been designed to be integrated with core courses and to become part of every day classroom life" (Yukon Department of Education, 2007). Some provinces and territories no longer have specific technology programs of study, or in some cases, technology is not listed with the other programs of studies, which may indicate that technology is meant to be integrated into other curricular areas and not treated as a separate area of study.

The requirement to meaningfully integrate computer technology into the curriculum and into students' educational experiences raises some questions and issues. Recent studies exploring computer use in primary and elementary classrooms critically examine how the computers are used and how students interact with the computers. Researchers, such as Norris, Sullivan, Poirot, and Soloway (2003) raise serious concerns about what they see as the failure to meaningfully integrate computer use into instruction in classrooms. In their words, "… although the literature points to the potential for impact, the reality is sobering: to a first-order approximation, the impact of computer technology over the past 25 years on primary and secondary education has been essentially zero" (p. 15). In essence, the curricular focus on meaningful integration has not been realized in practice. Several factors have influence over how computers are used in primary and elementary classrooms, and some of these will be discussed.

In selecting the literature for this review, professional studies and literature published since 1998 have been the primary focus, although earlier studies have been considered if they are still relevant to the contemporary primary or elementary classroom. Even over the course of a decade, however, many issues have changed in the consideration of how computers are used in the classroom or school, making it somewhat difficult to relate findings of earlier studies to the present day classroom context. The studies selected represent research conducted in many contexts, both Canadian and abroad, which examine how computers are used in classrooms, the issues that affect how they are used, and which type of software are used for assisting in the teaching of reading and writing to students in the primary or elementary grades. Although it was expected that there would be a multitude of studies in each area examined in the next sections, it was somewhat surprising to find perceived gaps in the treatment of computer use in primary classrooms, in particular.

Computer Use in the Primary Grades

Computers are present in almost all Canadian public schools however, Dwyer's (2007) study of elementary school computer use found that although teachers and administrators supported the use of computers by primary and elementary students, the way in which they allocated computer resources to and used computers with students varied by age. Dwyer found, for example, that the newest computers were being allocated to elementary classrooms, and that primary classrooms were receiving fewer and older computers. Although some primary teachers were concerned about how computers were being allocated in the schools, some teachers in the primary grades did not believe that it was appropriate to use computers with younger students, or believed that drill and practice language arts and math activities were most appropriate for these students. In addition, "… students generally used the computer once a week as part of reading groups.

Use of this software was repetitive, often the same software for a term" (p. 94) and several primary teachers were, "… not convinced that the computer developed the necessary literacy and mathematical skills students of this age group would need any more than traditional teaching methods" (p. 94). Privileging of elementary over primary grades for technology allocation and use was evident in several schools Dwyer studied.

Despite having computers to use in their classrooms and computer labs and in spite of all of the planning and money that has gone into acquiring computers for classrooms and schools, the computers are underutilized, and "… most students still spend most of their school day as if these tools and information resources had never been invented" (Becker,1998, p. 24). By Becker's rough estimate computer resources exist for students to spend about two hours or more per week using computers in school, yet most students reported approximate usage at one third of this time. When students in Becker's study were using computers, many of the computers were outdated as well as the software, so that students were not using newer applications like desktop publishing, online encyclopaedias, email or the internet, and were instead using computers most often for drill and practice activities, instructional games, or word processing in the primary and elementary grades. Rather than integrating computer use across all content-areas of the curriculum, in primary and elementary classrooms, "… the vast majority of computer time remains linked to skill instruction – primarily basic arithmetic and language arts skills in elementary schools" (p. 27).

Even though his findings are a little dated, Becker's (1988) commentary on the changes in thinking about how computers are utilized in schools is worthy of note. In the early 1980s the instructional focus was on teaching students to program computers using such programs as BASIC and Logo. This emphasis on programming shifted in the mid-1980s to drill and practice programs, or, in Becker's words, to "... use networked systems that individualize instruction and focus on increasing test scores" (1988, p. 25). Shortly afterwards, thinking shifted to embrace computers as tools, with an emphasis on word processing or on content-area specific programs such as databases or tutorials. In the early 1990s, the focus of instruction was multimedia programming, email, and then on Internet use towards the middle of the 1990s. Dwyer (2007) and Becker (1998) both argue that computer use in the primary and elementary grades is still very much based on notions and philosophies of computer use which date from the 1980s and 1990s, that being drill and practice programs, educational games, and word processing software. Both also argue that computers are being under-utilized, especially in the primary and elementary grades, in terms of time used and lack of attention to all of the potential functions of technology beyond drill and practice in skills, word processing, or games. Becker argues that the three main instructional functions for technology are (a) information transmission, (b) skill development, whether drill and practice or more complex skills such as problem solving skills and analytic thinking skills, and (c) communication between the classroom and the outside world. Becker advocates using computers for more complex skill development and for communication, rather than always using the computers for information exchange or simple skills development through drills.

Computer and Technology Integration in Primary Classrooms

What would computer integration look like in a primary classroom and how would student learning be affected? Page (2002) examined the effects of the significant integration of computers and other technology into primary and elementary classrooms by contrasting experimental groups which experienced significant amounts of daily integration of technology with control groups which experienced little or no technology use. While teachers in the control classrooms continued to teach in the traditional manner, teachers in the experimental classrooms received new resources and extensive training in technology use and integration. The experimental groups were provided with several computers with software such as *Microsoft Office*, *Hyper Studio*, *Kid Pix*, *Math Blaster*, *Grolier Multimedia Encyclopaedia*, and *Portfolio Assessment Toolkit* (pp. 397-398). A wide range of other technologies such as videoconferencing cameras, presentation screens, printers, and scanners, which are typically not present in regular classrooms, were provided to each of the experimental classrooms. The computers and other technologies were integrated throughout large portions of the daily, whether the curricular outcomes being addressed suggested technology use or not.

The imbalance in training and resources may have impacted the results of the study, but Page's findings (2002) are interesting in light of classroom practice and student achievement. Page noted that in the experimental classrooms, students communicated more with each other and initiated dialogue, whereas in the control classrooms, communication was primarily teacher-driven and teacher-initiated. He noted that students in the experimental classrooms seemed more confident with higher self-esteem than the students in the control classrooms. He also noted significant gains in mathematics, although his data with regards to language arts were inconclusive. Overall, Page noted very positive gains for students when technology was integrated in significant amounts of the curriculum and daily operations of the primary classrooms. In fact,

> … technology-enriched classrooms were far more likely to consist of a student-initiated environment where students participated in not only teacher-led instruction but also student instruction in the form of computer workgroups. … Technology-enriched classrooms were prone to produce more student-centered and individualized interactions, and non-technological classrooms consisted of the traditional model of teacher centeredness. (p. 403)

In Page's view, technology integration has significant positive impact on primary classrooms, yet the amount and type of integration which Page established in his study, and the amount and type of teacher training and resource allocation which occurred are rare in most primary classrooms.

Butzin (2001) helped develop and evaluate a computer and technology integration model where students took part in multiyear, multi-grade groupings in which technology was integrated into reading, writing, and mathematics in a non-traditional classroom setting. Butzin remarked that traditionally "most teachers still have difficulty integrating computers into classroom instruction. …As a result, teachers tend to use computers as an "extra" for students who finish their written work or who need supplemental practice" (p. 372). In the model developed by Butzin, students interact meaningfully with technology for a minimum of one hour during every day as a direct support for their coursework in reading, writing, and mathematics. In Butzin's study, students completing the three year cycle of programming scored significantly higher on reading, writing, and mathematics tests than students in the control school who, although they had similar access to technology, relied on traditional teaching methods to teach reading, writing, and mathematics. Her conclusion, based on the data collected, is that the number of computers in classrooms should not be the focus of the technology debate. Instead, the main issue is how computers are used, and whether the instructional models in traditional classrooms support or hinder meaningful technology integration.

Issues Affecting Computer Use in the Primary Grades

Why are computers not being integrated into daily classroom practice to the extent that Beckett (1988), Butzin (2001), Page (2002) or Norris, Sullivan, Poirot, and Soloway (2003) would intend? Many studies examine the issues affecting how computers are used in classrooms. Two of the major issues identified in several studies were availability of resources, which encompasses the number and type of computers available, their location within the school and issues of scheduling, and teacher attitudes towards and preparedness for teaching using computer technology.

Availability of Resources

Computers are used in a variety of ways in primary schools and classrooms and every school allocates its technology resources differently. Judge, Puckett, and Bell (2006) report, for example, that lower income areas tend to have a slightly lower ratio of children to computers (approximately five to one), but that these computers tend to be found predominantly in central labs, rather than in classrooms, and that these computers are used for drill and practice activities approximately thirty-five percent of the time. In contrast, schools in higher income areas tend to have a higher student to computer ratio (approximately six to one), but more computers are located in classrooms, are used less frequently for drill and practice, and more often for other applications like internet use. The physical location of computers within the school impacts how they are accessed and used by teachers and students. For example, if computers are found in a central computer lab, there will be more computers available for use at one time, but teachers must negotiate lab times, which are often at a premium, so time on the computers is more limited. If computers are found in classrooms, the potential exists for more frequent or spontaneous use of the computer by students, but students will have to share access, and often, as Dwyer (2007) reported, classroom computers allocated to primary and elementary classrooms tend to be older with outdated software and capabilities.

Having computers available for use and able to be accessed by students and teachers is the first step to integrating computer use into the curriculum. If resources are unavailable, meaningful technology integration into other curricular areas is unfeasible. Norris, Sullivan, Poirot, and Soloway (2003) report that up to twenty-nine percent of teachers do not have adequate access to technology for student use in their schools. In these teachers' views, students do not have enough access to learn to use computers effectively and certainly not often enough access for teachers to adequately integrate computers into the daily education of their students. Simply having one computer for every five or six students in the school does not ensure that the students actually have meaningful access to computers. Computer labs and classroom computers may be underutilized. Restricted computer access and scheduling, as well as the unequal distribution of computers by age and quality of computer are deterrents from the amount and type of computer use by students.

Teacher Preparedness and Perception

Dwyer's (2007) research, as well as Wozney, Venkatesh, and Abrami's study (2006), both list teacher preparation as a significant factor affecting how often teachers use computers with the students in their classrooms and the level of integration with which they are comfortable. In Dwyer's study, especially, she found that teachers in the primary grades felt less prepared and confident than their colleagues in the elementary grades to use computers with their students. Franklin (2007) found that even when teachers felt sufficiently prepared to use computers in the classroom, and believed in constructivist pedagogies, that computers were still being used mainly for drill and practice activities in the primary grades, and were not used as frequently by primary students as by elementary students.

Wozney, Venkatesh, and Abrami's Canadian study (2006) seems to dispute the assertion that teachers mainly use computers for drill and practice, although the study does not distinguish between the primary and elementary grades. According to Wozney and her colleagues, less than half of the surveyed teachers reported using computers for drill, practice, tutorials or remediation, although most did not use computers for complex applications. "Computers may simply maintain existing instructional practices that traditionally focus more on transmitting information than helping learners actively construct knowledge" (Wozney, et al., 2006, p. 193), and several teachers reported that although they were shown some interesting applications on the computer, they did not feel they were shown how to use computers with students in practical or useful ways. In one teacher's words,

> I would like to learn an application that I need and my students need. I want to use what I learn. It is fine to know how to take a photo and make a book or calendar but is that truly what a class computer should be used for? What are the things students will need to know in the future? (p. 194)

This teacher's questions highlight a significant problem in teacher technology training and the philosophy behind technology use in schools. While the expectation is that technology be integrated meaningfully into the curriculum and used to complement classroom instruction in new and effective ways, the teachers reported that they were shown interesting new applications that were neither practical nor useful to enhance instruction in other curricular areas.

Planning and implementation time were major factors in Bauer and Kenton's (2005) study of obstacles to technology integration in schools. They found that although teachers might have training and be confident with the use of computers, there were several barriers to the meaningful integration of computers into the curriculum and these barriers had strong implications for how often and how effectively students in all grades experienced computer use in school. All of the teachers in Bauer and Kenton's study were considered to be technologically proficient, yet they were not integrating computers into their teaching. Several of the teachers complained that the switch from traditional lesson delivery in a classroom to integrated instruction with technology required too much extra planning time, and also backup plans needed to be made in case of technical difficulties. The need to share computers between students in the classroom also made keeping students on task a frustration for teachers. For these reasons, even teachers who were adept at and confident with computer use, who were committed to trying new and innovative lessons to integrate computer use into the curriculum and had adequate access to computers in their schools and classrooms were not integrating them effectively within the curriculum and their students were not experiencing the effective use of technology in their instruction.

On the continuum of integration, from “familiarization” to “ utilization” to “integration” to “reorientation” and “evolution”, which Bauer and Kenton (2005) described, most teachers achieved the utilization stage only. The familiarization and utilization stages, during which teachers first are aware that computers can be used to complement instruction and then superficially add computers into their teaching, are the lowest levels of effective computer use in teaching. In the utilization stage especially, teachers “… become prematurely satisfied with their limited use of technology, but lack a positive commitment to it and readily discard the technology at the first sign of trouble” (p. 522). These teachers may be satisfied that they use computers with their students, but they are not choosing to integrate computer use meaningfully. Bauer and Kenton argue that the real change in instruction happens at the integration phase and beyond, where teachers first purposefully plan for and rely upon technology for appropriate curricular outcomes, then change teaching practices to more effectively integrate technology. However, since most teachers, even the technologically proficient teachers from Bauer and Kenton’s study, never progressed past utilization, where they treat computers as an addition to the program, not as a vital part of the program, then the use of computers is not as deliberate or effective as it could be for teachers who actually integrate technology instruction into the classroom and into all curricular areas.

Why is teacher preparation and perception important to the discussion of how computers are used in primary classrooms? Simply, despite what the curriculum may say, and despite issues of availability and resource allocation, the first decisions about when and where technology is used in the curriculum is dependent upon teachers. When teachers are comfortable with technology use and see the value of it, then they will use computers more meaningfully with their students to support their learning across the curriculum and integrate them into the curriculum. Bauer and Kenton contrast computer integration, or "… the full-time, daily operation within lessons" (2005, p. 535) with simple computer use, noting that most respondents indicated they used computer technology less than fifty percent of the time in delivering lessons. The vast differences in outlook and use of computers by teachers indicate that the way in which the students experience computer use in their schooling also varies widely, regardless of the stated curriculum expectations.

Types of Available Software for Teaching Reading and Writing

Keeping in mind that teachers and students have varying degrees of access to computers in their classrooms and schools, it becomes important to look at the different ways software can be used to assist reading and writing development. Many studies deal with specific software programs or types of software and their effects upon student achievement in reading and writing, particularly for struggling readers and writers. Software designed to assist early or struggling readers and writers to improve their reading skills typically targets one or more specific areas of focus related to reading or writing and then addresses those skills through direct instruction, demonstration, and student practice. Often, there is an evaluative component to the software as well. The ability to evaluate student progress or needs and adjust the content of the program accordingly is imperative for educational software in a classroom setting (Bishop & Santoro, 2006) whether the software is presented as a game, an interactive program, or any other type of software platform. Although the manner in which content is presented differs between software, without the evaluative component, teachers are unable to make decisions about when to move on to more advanced skills.

The studies addressed below are by no means a comprehensive list of all of the software for teaching or assisting in teaching reading and writing skills, however they are representative of the types of software that are available, and some of the skill areas that are addressed by software.

Reading Skills Which Are Addressed by Software

Phonological awareness. The teaching of phonological awareness, or the awareness of sounds in language is especially suited to the multimedia capabilities of modern computers. For example, software can be designed to use speech feedback that highlights a word's parts as they are being spoken by the program (Fasting & Lyster, 2005) or to present storybooks with speech, other multimedia features that highlight the skills being worked upon by students (Littleton, Wood & Chera, 2006). In teaching phonological awareness, sounds should first be presented orally by the computer with concrete representations and should allow for manipulation of the sounds by students before being linked to letters (Bishop & Santoro, 2006).

Littleton, Wood, and Chera (2006) evaluated *Talking Books*, which are programs designed to promote phonological awareness by having children select whether the program would read the entire passage, parts of the passage, or none of the passage to them. Chera had previously developed the software in 2000, which was then used in the 2006 study. In this study, five and six-year-old boys were given a pretest composed of batteries of formal, standardized tests and informal tests created by the authors to evaluate phonological awareness. No control group was indicated in the article. Once testing was completed, the boys were given a book to read on the computer with *Talking Books*. The boys then used their comfort in reading the story to determine the level of support offered by the software program. Confident readers were able to read the story with no assistance from the computer and no use of the computerized speech. Less confident readers were able to have the computer read parts of the story, or the entire story while the students followed along on the screen and read the story. The researchers found that children who used the programs to read to them, or who read with the support of the program made significant gains in their phonological awareness in the areas of rhyme detection, and alliteration detection and production, and also showed increased confidence while reading.

Fasting and Lyster (2005) created their own program, *MultiFunk*, to teach phonological and phonemic awareness to Norwegian children through the similar use of computer-generated speech to read all or part of the passages to students. Grades five, six and seven students were recommended for the study by their teachers or reading therapists and were assigned to the experimental or control groups. The authors administered vocabulary, oral single word-decoding, silent sentence-reading, and spelling tests to the students in both the experimental and control groups to establish baseline data for the study. During the course of the intervention, which lasted for seven weeks, the students chose texts to read which were adapted for computerized voice and text-to-speech capabilities. A supplementary writing component was available, but not required. Although spelling was assessed in both pre- and posttests, spelling was not formally addressed in *MultiFunk*, instead, students read aloud from passages and corrected them. In each test, the experimental group tested lower initially than the control group, but tested at or above the level of the control group on the posttests. Males in the experimental group showed greater significant gains in spelling than all other groups, for example, although all students in the experimental groups showed more gains than the students in the control groups even in cases where students from control and experimental groups were in the same classrooms. Within the experimental group, the authors found that the computerized speech was a significant factor for improvement in the word and sentence reading tasks. In addition, use of the speech functions of the program allowed students to progress through tasks more quickly than students who did not use the computerized speech.

Many other programs also deal with phonological or phonemic awareness in different ways, such as through drill and practice activities. Often these programs are designed to stand-alone or supplement reading instruction by focussing on specific sounds. Neither the *Talking Books* (Littleton, Wood, & Chera, 2006) nor *MultiFunk* (Fasting & Lyster, 2005) appear to be as basic as the software that Bishop and Santoro (2006) discussed. Bishop and Santoro's evaluated programs which begin with concrete representations of sounds in isolation to be manipulated by learners and then progress on to reading words. According to Bishop and Santoro's criteria, programs teaching phonological awareness should begin with concrete manipulations of sound units, before introducing letters and words, although Littleton, Wood and Chera (2006) and Fasting and Lyster (2005) make no mention of doing so. Software programs designed for older children who are already reading may not follow Bishop and Santoro's criteria, although simpler programs designed for younger, less advanced users may follow Bishop and Santoro's criteria more closely.

Letters and sounds. Connell and Witt (2004) used computer speech capabilities to teach letter names and sounds to two kindergarten students in a more drill-based program. The students listened while the computerized speech read the names of the letters, and then clicked on the appropriate letter presented on the screen. After each child matched the names with the upper and lower case letters, the students matched the upper and lower case letters to their counterparts before matching words read by the computer to those printed on the screen. The program which Connell and Witt used for their study was not designed for classroom use, instead it was designed as an intervention to teach skills for struggling pre-readers. The program was able to be customized for different types of skills or activities, but still dependent on drill and practice type activities, where the student clicked on the answer matching the sound stimulus made by the computerized voice or the visual stimulus on the screen. This focus on computer-generated question, and then clicking on the correct response is a characteristic of drill and practice programs. The questions are determined in advance, and there is one correct answer to be selected for each question. Programs to teach letter and sound recognition or matching could conceivably follow other formats, however no examples were found, and drill and practice programs seem to dominate in this area.

Vocabulary development. The development of vocabulary and knowledge of word parts is an important part of the reading process. One such program that deals with vocabulary development and word parts is *Kidspiration*, a software resource which is used in several Canadian provinces and territories, such as Nova Scotia and the Yukon Territory (see Appendices A and B).

Programs such as *Kidspiration* (Gill, 2007) use word parts to teach vocabulary. *Kidspiration* uses webs and pictures to teach new vocabulary, but is restricted to text and pictures with no speech. For example, Gill chose "tri" and then created a web that included "triangle" and "triplets" (p. 82). *Kidspiration* uses graphical organizers, pictures, and definitions to reinforce vocabulary development and gives students the option to choose their own words and word parts based on interest. Programs similar to *Kidspiration* help students to organize words and images, and make connections between words. Unlike a drill and practice based vocabulary or phonics program, *Kidspiration* supplements classroom teaching of specific skills, such as vocabulary development in this instance, by student choice and exploration. In this type of program, there is no preset question and no set of correct answers. Exploration and problem-solving are used instead of repeated drills. Vocabulary and word parts could be, and in fact often are, taught by drill and practice programs, yet this is not the case with *Kidspiration.* Unfortunately, although she explains and demonstrates the functions of *Kidspiration,* Gill does not provide an evaluation of the efficacy of *Kidspiration* for use with students or provide evidence of its effects on student vocabulary development.

Reading fluency and comprehension. Sorrell, Bell and McCallum (2007) evaluated a program designed to improve reading fluency for struggling readers. *Kurzweil 3000*, which is approved for special needs students in Prince Edward Island (see Appendices A and B), is a text-to-speech reading program. In this study, the researchers evaluated the reading rate, fluency, and comprehension of elementary age students who used the *Kurzweil 3000* software and those who did not. *Kurzweil 3000* and other text-to-speech reading programs "… are similar to traditional classroom reading methods such as guided reading in that they include an auditory component with a visual representation of the text" (p. 2) and allow students to read text that may be too difficult for them to decode independently. The software program presents passages to the student and the student reads along with the computerized voice. Results show that this program was effective in improving the reading rate and fluency for students reading below normal reading rate for their age, but that *Kurzweil 3000* forced faster readers to slow their reading rates, which negatively influenced their comprehension scores. It may be that these students were forced to slow down and concentrate on matching the pace of the computer-generated speech, not on comprehending what they were reading. It is important to note that the authors provided no account for why the more confident readers suffered decreased performance on the comprehension measures after using the *Kurzweil 3000* software. As the purpose behind using this program is to scaffold slower, less confident readers by offering the support of the computer-generated speech so they could hear how the passage should be read while reading along on the screen at a normal reading pace, it seems consistent that the program ought to benefit the slower, less confident readers. Students who read more quickly and confidently would not require this support and, as this study shows, regressed in reading comprehension.

Sorrell, Bell and McCallum (2007) used *Accelerated Reader* software to assess the comprehension of students who used the *Kurzweil 3000* software and those who did not and then compared their results on the comprehension assessments. In *Accelerated*

Reader, students read a trade book, whether fiction or non-fiction, and then take a test to evaluate their comprehension of the book. The trade books used for *Accelerated Reader* are levelled by difficulty, and each is assigned a point value based on this difficulty rating. Students take the multiple-choice test after reading the book, and answer five or ten literal comprehension questions from memory. No pictures or other types of prompts are available during the test, only the question stem and four responses. *Accelerated Reader* would be considered a proprietary program because, although it is not linked specifically to resources published by the software company, it is linked to the resources selected by the company and is unusable with books that are not on the list, or as a stand-alone resource without books.

Using *Accelerated Reader* and other measures of comprehension assessment, Sorrell, Bell and McCallum (2007) found that reading comprehension scores for faster, more confident readers who used the *Kurzweil 3000* program were negatively impacted when compared to their peers who did not use the software, whereas the comprehension rates of slower readers who used the *Kurzweil 3000* program were approximately the same as their peers who did not use the software. The authors did not control for reading level of the materials used for assessing reading comprehension using *Accelerated Reader* since all readers were at different levels, which may have impacted the results of the study. The decrease in the comprehension rate of the faster readers and the lack of significant increase in comprehension rate of the slower readers after using the *Kurzweil 3000* software was not thoroughly explained. It seems reasonable, however, in my opinion, to conclude that use of the *Kurzweil 3000* software would be more appropriate with slower readers, who increased their reading rates while maintaining their comprehension levels, than with faster, more confident readers, who suffered decreased comprehension levels after using the program because the program slowed them down too much.

The results of Sorrell, Bell and McCallum's research (2007) highlight the need to make purposeful decisions about when to use software, and also underscore the point that not all software is appropriate for use with all students. The authors concluded, "Many educators regard technology in the classroom as an innovative educational necessity. ...Even though schools cannot avoid the technological revolution, educators must become aware of which computer programs are supported through research as being both instructionally efficient and effective" (p. 11).

Writing Skills Which Are Addressed by Software

The writing process. Programs which are designed to help teach the writing process and skills seem designed more to scaffold by demonstrating writing skills and processes for young writers to follow rather than relying on drill and practice games. For example, the *TELE-Web* software that Englert, Manalo and Zhao (2004) reviewed helped young writers by demonstrating text structures for narratives and the *Summary Street* software that Wade-Stein and Kintsch (2004) reviewed reinforced writing skills through providing content and allowing students to summarize it, and by demonstrating the writing cycle. *TELE-Web* offers assistance with structuring writing by providing spaces where students write their topic sentences, supporting sentences and concluding sentences, but allows students to write on their own topic or a topic provided by their teacher. *Summary Street* provides less assistance with structure than *TELE-Web*, but more assistance with content, as students summarize pre-existing text, such as an essay on the solar system, using their own words with the length of the summary being monitored by the computer for all sections so that it is balanced and covers all sections adequately. Both of these programs are designed around the concept of scaffolding writers' development by offering variable amounts of assistance, based upon need.

The main concept of the *TELE-Web* software is to scaffold young writers "...by controlling or prompting those elements that are beyond children's range of initial competence while permitting them to concentrate on the writing aspects that are within their grasp" (Englert, Manalo & Zhao, 2004, p. 7). In this study, the *TELE-Web* software assisted first and second grade students to compose short narrative paragraphs by prompting students to write one introductory sentence, three supporting sentences, and a concluding sentence. The program then put the sentences together into a traditional paragraph structure without line breaks between the sentences. As part of the program, students used *TELE-Web* to write narratives using both the full support with extended instructions and spots for each separate sentence and the unsupported mode with just an area in which students wrote free-form, as well as writing narratives without the benefit of the program with paper and pencil. The narratives created using both modes of *TELE-Web* as well as without using the software were scored, and the full support mode of the *TELE-Web* software was determined to be the experimental condition in which most students scored highest for content. The researchers concluded that one of the main features that makes *TELE-Web* so successful is the adaptability that allows teachers to program more or less assistance to meet the varying needs of students. In fact, some students even transferred the organizational elements of *TELE-Web* into their writing when they were not on the computer, such as the naming conventions for each sentence.

In the example cited in the study, one student wrote "toppec seitse" (topic sentence), "sentse 1" (sentence 1) (p. 15, Figure 4), and so forth, beside each sentence that he wrote for his *paper-and-pencil* narrative, which emulates how *TELE-Web* names each sentence in the supported paragraph function. Following the supported paragraph structure of *TELE-Web*, the student wrote one sentence per line, rather than putting all sentences together into a paragraph format, although he was able to create a narrative with a main idea and supporting details, and appropriate capitalization and punctuation. The authors praised *TELE-Web* for helping this student learn to structure his writing, stating, "Tyler's behaviour suggested that the TELE-Web prompts and scaffolds were consciously retrieved to mediate his thought and performance during text composition, even though he was composing text using paper and pencil" (p. 15). It is interesting to note, however, that Tyler learned to add unnecessary details in his writing and thus may have written less as a consequence.

Summary Street, a program for teaching writing skills and reading comprehension to elementary or older students through summarization, was the focus of the review by Wade-Stein and Kintsch (2004). The ability to succinctly summarize expository text using expository structure, even during the elementary years, helps students both understand expository structures when they encounter them while reading, and prepares them for writing using expository text structures as well. During a three-year piloting of *Summary Street* in two grade six classrooms, the authors collected data from students' interactions with the program and adapted it to be more user-friendly and pedagogically sound. Wade-Stein and Kintsch suggest that the program design of *Summary Street* guides students to know how much to write, and how to segment an account so that all important parts are summarized with enough depth.

The first criterion *Summary Street* uses to determine whether the passage has been properly summarized is length. Each section is assigned criteria for determining the appropriate length of summary and when students submit their paragraph for the section they are summarizing, a feedback window appears which uses a gauge to show them whether the summary is too short, approximately the right length, or too long. The actual content of the sentences is not ascertained to see how closely they summarize the passage. Next, *Summary Street* checks for redundancy and relevancy of details by examining each sentence and comparing the wording to other sentences, prompting students when wording is redundant, or adding suggestions for improvement, such as "Note: Spelling errors can cause me to think good sentences are irrelevant. Be sure to check your spelling!" (Wade-Stein & Kintsch, 2004, p. 344, Figure 6) when the program encounters a spelling error. At this point, students are able to add or delete words and sentences to correct for redundant wording, or improve relevancy. The authors found, however, that the students tended to concentrate on adding or deleting words or sentences, without regard to maintaining passage consistency or meaning. Wade-Stein and Kintsch also found that when a greater degree of feedback was given to students, their summaries tended to be of better quality than when given less feedback by the software. For example, when feedback was given for length only, quality indicators like redundancy or relevancy were ignored.

Software Features that Assist in Teaching Reading and Writing

Many newer software programs take advantage of the unique multimedia capabilities of computers, such as speech and graphics, are interactive and adaptive to student needs, and have the ability to allow students to choose the level of support they are comfortable with when being assisted with reading or writing. Programs such as those evaluated by Littleton, Wood and Chera (2006), Gill (2007), Englert, Manolo and Zhao (2004), Wade-Stein and Kintsch (2004), or Mostow, Aist, Burkhead, Corbett, Cuneo, Eitelman, et al. (2003) specifically use these capabilities to reinforce and scaffold the learning of reading or writing skills for children. One of the main features used is computerized speech. Software uses speech to read to students or with them, to give sounds of letters so that students can match sounds to letters, and many other uses, dependant on the software design. Pictures allow students to see concrete examples of sounds or words they are reading or to incorporate images and text into organizational charts. In addition to multimedia, another feature that makes programs effective for assisting in reading and writing appears to be adaptability, where teachers or students can select how much assistance is required for the task, and adjust for the needs of each student. Finally, as Bishop and Santoro (2006) write, educational software should have an evaluative component and provide feedback to the student and the teacher so that it can be adapted to student needs and student progress can be monitored.

Methods of Software Evaluation

For the purposes of my study, it is important to discover what types of software are available and authorized for use in assisting in the teaching of reading and writing in Canada and then find a meaningful way to evaluate that software. Several recent studies have offered possible evaluation methods for assessing software. Often in these studies, the software itself is not being evaluated, rather student performances of the targeted skills while using the software are typically compared to either their own performances while not using the software, or to the performance of a control group.

Evaluating Software Through Experimental Designs

Experimental studies, such as Littleton, Wood and Chera's (2006) study were designed to evaluate specific software packages using pre- and posttest designs. The categories tested in their study were literacy based, and focused on testing phonological awareness growth for at-risk students who used talking books software. Fasting and Lyster (2005) developed their own software for teaching reading and spelling to at-risk students and then assessed the effectiveness of the program using a pretest, posttest, control group experimental design to test students' reading and spelling performance on standardized tests after using the program. Neither study tested the software itself, instead, both focussed on student growth in the targeted skill areas. Mostow, Aist, Burkhead, Corbett, Cuneo, Eitelman, et al.'s (2003) evaluation of a reading tutoring program, or Lefever-Davis and Pearman's (2005) evaluation of CD-ROM storybooks follow a similar pattern with experimental designs which track student growth after using the software, or compare student performance using the software against performance without the software. Although these studies offer insight into how software can assist in skill development for reading, this approach is not appropriate for the research questions guiding this thesis because they examine only one software program, instead of several programs, and they do not examine publishers' claims about what their products do. Evaluating each software program using an experimental design would be impractical.

Evaluating Software Through Performance Measures

Performance based measurements are also being used to test how students perform on tasks after using or while using the targeted software. Englert, Manolo and Zhao (2004) used a rubric to evaluate student writing using *TELE-Web* software. Their performance-based assessment of the students' writing used the categories of productivity, conventions, and genre-related features and compared pieces written both with and without the assistance of the software. Again, the evaluation criteria focussed exclusively on writing performance, not the software itself. Wade-Stein and Kintsch's (2004) review of *Summary Street*, a program designed to teach writing skills through summarizing, was another example of a study designed to evaluate the literacy growth of students using specific programs designed to improve aspects of reading and writing performance. The written work produced by students in both experimental conditions was scored, and the use of the software was found to be beneficial. Time on task, content and quality of summaries, and response to program feedback were the main evaluation criteria used. As in the previous examples, these studies focussed on the effects of software use on student performance in literacy tasks and not on the software itself, whereas the scope of my thesis is to survey several software programs, and to not examine levels of literacy growth in specific students using each program.

Evaluating Software Using Student Perceptions

A third type of evaluation strategy that looks beyond the educational capabilities of the software into usability criteria and enjoyment is evident in Sim, MacFarlane and Read's (2006) study of three software programs designed to prepare seven- and eight-year-old students for a national standardized test in science. In Sim and colleagues' study, the children rated each software program on an ordinal scale on the broad categories of fun, usability and learning, and then the researchers observed the children and took anecdotal notes to evaluate fun and usability as well as using pre- and posttest measures to evaluate student learning on each program. Although this study assessed software design and utility along with learning outcomes, the categories are very broad and subjective.

Evaluating Software Design and Content

After investigating various evaluation strategies for software, whether experimental with pre- and posttests, performance based, or subjective measures of preference, it was determined that none of these strategies is an appropriate method for answering the research questions. Either the criteria were designed specifically for the program studied, focussed on literacy outcomes to the exclusion of program design and usability, or were based on student preference, rather than utility for teachers. In each case, there was no mention of evaluating the claims made by software manufacturers against how their software actually performed. Software that scored highly on program design, for example, might not teach what it claimed to teach in a pedagogically sound manner. It might be engaging and stimulating with good graphics and motivating activities, yet it might not fit with the way reading and writing is taught in schools. On the other side of the spectrum, a program might deliver content in a pedagogically sound manner, but might not have a good interface design. No single criterion can indicate whether a software program is a good program to use with students, nor does scoring higher or lower on any of the criteria. The researchers designing the evaluation criteria address their own concerns and research questions. Every classroom context and every student's needs are different, so good evaluation criteria need to be flexible enough for teachers to make decisions about which programs will meet the specific needs of their students. For example, *Kurzweil 3000*, a text-to-speech program, might be appropriate for use with certain students, but not with more competent readers (Sorrell, Bell & McCallum, 2007). Utility, pedagogical practice, and program design are my primary concerns as a classroom teacher, and inform my choice of the criteria which define good programs. As a way to evaluate a program for first use, I would not perform a full experimental study, such as those described previously. Instead, I would look for criteria to indicate how useful a program is. Once I had used the program, and wanted to evaluate its efficacy for assisting in the teaching of reading and writing, then I would perform a quantitative study, similar to those described above. This was my rationale for rejecting an experimental or performance-based assessment at this time.

In contrast to the assessment criteria based on student performance or interest, two examples of rubrics were found that assess utility for teachers, and program design. McVee and Dickson's (2002) rubric is short, and somewhat general and subjective, but would work well for classroom teachers who are not concerned as closely with the mechanics of program or interface design. McVee and Dickson chose seven criteria based on presentation, ease of navigation, flexibility and adaptability, assessment functions, meeting student needs and interests, value, and alignment with instructional goals and teaching philosophies. They discussed, at length, the process of determining evaluation categories based on the needs of the teacher, and then presented their results for *WiggleWorks* software, which scored the highest of all the programs they evaluated, with a final score of 19 out of 21 total possible points. *WiggleWorks* software is designed for children in kindergarten and the first three grades and provides texts for students to read either on the computer screen or a hard copy, allows students to record themselves reading, and offers activities related to the stories being read. McVee and Dickson's rubric seems suitable for the purposes of this research, but the categories are still somewhat vague, and the actual descriptors for choosing between a score of low and a score of medium or high are not provided.

Bishop and Santoro (2006), a software designer and a beginning reading instructor, developed a thorough framework to evaluate various aspects of interface design, content, and instructional design of the software and also suggest a specific approach for testing educational software (see Appendix C) which includes testing how easily students could use the program interface, what instructions and support are available, and examining whether the way the software program presents its content is pedagogically sound. Unlike the experimental designs or the performance measures discussed previously, the authors did not test student literacy learning using specific programs, although they did include indicators of good pedagogical practice which should, in the authors' view, promote reading development.

Using their criteria, Bishop and Santoro (2006) found that software which addressed reading and writing skills in more pedagogically sound ways was often not engaging for students and showed poor overall design and use of multimedia content. In contrast, software designed to stimulate student interest or with better interface design "… include very little direct teaching or instructional support to accompany various exploratory activities. Furthermore, very few titles from this category address phonological awareness skills at any level" (p. 68). Bishop and Santoro suggest that although few titles score highly on program design, content, and delivery, teachers could supply additional support to their students to make the software more beneficial.

Bishop and Santoro's (2006) rubric is based on three categories with various subcategories and indicators related to each broader category. Under *Interface Design*, the authors chose *Aesthetically Pleasing*, *Supportive Operationally* and *Interactive* as the criteria for evaluating how well the program was designed. *Aesthetically Pleasing* and *Supportive Operationally* would fit my third research question on appropriateness for the audience. If the software program is well organized, interesting, and adaptable, as well as easy to use, with both written and auditory instructions, and provides feedback and support to the learner, then the program may be appropriate for student use. The *Interactive* criterion, dealing with how passive or active the student is in the experience and how responsive the program is to student input would fit under the fifth question, "What benefits do students gain and do they actually learn from these software programs"?

Bishop and Santoro's second category (2006), *Content Criteria*, covered the reading outcomes identified as important by the authors for emergent readers. The *Phonological Awareness* criterion addressed sound and concrete representations, claiming that instead of pairing phonological awareness with phonics or letter recognition, software should first focus on sounds without text. The second criterion, *Alphabetic Understanding* dealt with letter-sound correspondence and word reading. These categories are not useful for answering broader questions about reading and writing development, and should be expanded in order to cover more relevant features of reading such as vocabulary development, reading fluency, and comprehension, and writing such as spelling, idea generation, and content, however the authors were focussing primarily on emergent readers.

The final broad category in Bishop and Santoro's research (2006) is *Instructional Design* and included criteria related to the teaching methodology used by the software programs they evaluated. The criterion *Systematic* was used to evaluate how consistent the program was in its delivery of content, and whether it followed good teaching practice through informing students of objectives, providing examples of new skills, and providing opportunities for students to practice and master skills in a sequential manner. *Instructionally Supportive* evaluated how much support and freedom students were given in the learning experience and how prescriptive the program was. In *Assessing*, the researchers looked at how the programs assessed student work, how they used their assessments to guide or adapt instruction, and how the programs tracked student progress in ways the student and teacher could access and use. The final criterion of *Motivating* assessed how the program made learning relevant to the learner, how it built confidence through challenges and successes, and the reward system offered by the program. These criteria addressed my third and fifth research questions, suggesting that Bishop and Santoro's criteria were applicable, with some modification, for the majority of the research questions posed in my study.

CHAPTER 3: METHODOLOGY

In examining the effectiveness of the authorized software approved as core and supplementary resources for the teaching of reading and writing in Canada, several steps were considered. The research and professional literature regarding computer-assisted instruction and the issues facing schools for the use of computers were examined. Next, these issues were synthesized to show the value and purpose of evaluating the approved software programs, and then to create a plan and method to systematically evaluate the software. The results indicated the typical patterns of what is available for authorized resources for the teaching of reading and writing through computer-assisted instruction in Canada.

Software Program Selection

The main materials required for this thesis are copies of the approved software for each of the provinces and territories in Canada. As every province and territory has control over its own curriculum and resources, each jurisdiction has different software resources approved for teaching reading and writing to students in the primary grades.

The first step in procuring materials for this thesis was accessing the website for each provincial and territorial Ministry of Education, and then sending an email to each department requesting the list of software resources. As all of the software for the other provinces is in English, the Province of Québec and the Territory of Nunavut were excluded from this study, because resources were primarily available in French or Inuktitut. The list of resources, as ascertained by personal communication, through browsing websites and on-line catalogues is found in Appendix A. Many of the software titles were published prior to 1998, making them more than ten years old. Most provinces and territories have different lists, however there is some overlap. Alberta, British Columbia, Saskatchewan, Manitoba, The Northwest Territories and the Yukon Territory have extensive overlap because all have signed the Western Canadian Protocol Document and maintain its software listings on their list of resources (Western and Northern Canadian Protocol, 2008), even though the software is dated. *Chicka Chicka Boom Boom* (1996), *Easy Book Deluxe* (1998), *Kid Works Deluxe* (1995), and *Has Anybody Seen My Umbrella?* (1998) are the listed resources under English Language Arts.

After the list of provincial and territorial software resources was established, the publishers of each resource were contacted via email or fax and an evaluation copy of each title was requested. In some cases, such as for *Kidspiration 2*, *Inspiration 8* and *SMART Ideas 5.0*, downloadable trial versions were available. One copy of *Easybook Deluxe* was sent as an evaluation copy by the publisher, and the rest of the programs evaluated were purchased from the publishers or their local distributors, and other online sites, especially since several of the programs listed are discontinued and no longer distributed by their publishers, such as *Has Anybody Seen My Umbrella*.

Unit of Analysis

All software programs from the list of resources for each province were compiled into a general list and itemized by province or territory. Although there was some overlap, due in part to the Western Canadian Protocol documents, most provinces and territories had different programs. Each software program contains material on CD-ROM, although some are available for either Windows or Macintosh platforms. Support materials varied from printed teachers' manuals to electronic versions found on the CD-ROMs themselves. The majority of the programs were available for Windows platforms, and these were analyzed. Several programs were no longer in production by their publishers, although some were available and purchased from at various on-line locations.

Development of Evaluation Framework

To develop the evaluation framework, various studies were examined, such as Littleton, Wood and Chera's (2006) experimental design, Englert, Manolo and Zhao's (2004) performance-based rubric, and Sim, MacFarlane and Read's (2006) study of student perceptions of learning and fun while using software. The experimental designs were inappropriate for this thesis, as they tested one or few software programs, and did not examine program claims. The performance-based criteria also tested one or few software programs, and tested student performance, not the workings of the software itself. Broader criteria were needed, similar to McVee and Dickson's (2002) rubric, to evaluate various software programs for utility in the classroom and program design, however the broad, undefined categories were problematic because the authors did not provide indicators for each criterion, nor an indication of what would be considered a high score, a medium score, or a low score. For example, McVee and Dickson's criterion stated, "What observations can be made about overall media presentation" (p. 637) without giving indications as to what the evaluator should consider high quality media presentation. Bishop and Santoro (2006) provide a similar criterion, but their indicators guide evaluators as to what constitutes a good presentation. In addition to the criterion, "To what extent is the software… aesthetically pleasing" (p. 59, Figure 1), their indicators (p. 59, Figure 1):

> The media used is high quality. Screens are laid out in well-organized ways (rather than haphazard placement of objects). Screens are neither overly stimulating nor boring. The "look and feel" of this program is likely to be pleasing to the learner. Media are used to create themes/metaphors that relate to the content and help create meaning.
>
> Learner is able to modify the interface according to individual preferences.

provide specific evaluation guidelines for evaluators to follow. It was concluded that Bishop and Santoro's (2006) categories were most similar in purpose to my thesis, and that professionals in both the reading and teaching field, and software design field developed the evaluation criteria, so these categories were adopted, and most of the indicators, where appropriate. Where they were not deemed appropriate, the indicators were changed to reflect the research questions.

The main area of change in the original criteria was to the content category. The program content criteria were modified to encompass a larger selection of software than in Bishop and Santoro's study (2006), which looked only at phonological awareness and alphabetical understanding. For the purposes of my thesis, Bishop and Santoro's categories related to reading and writing skills were replaced by using the documentation accompanying the software. The categories of reading or writing skills which the documentation claims are addressed were noted, and then the skills which the programs claim are taught were used as indicators to determine whether the program met or did not meet the indicators for the criteria. Where two software programs overlapped in their content areas, the indicators were compared and analyzed to see whether they were consistent across programs. In this manner, the new categories, which are similar in format to Bishop and Santoro's categories, were used to address my second and fifth research questions. Bishop and Santoro also provided an excellent procedure for testing software features, which was followed as closely as possible (see Appendix C).

The software programs were coded according to their function based on their accompanying literature. Where functions were identified that were not related to reading or writing skills such as drawing, science, the humanities, or mathematics, they were ignored for the purposes of this study. In some cases, the software had more than one main function, and in these cases, the two major or two most different functions were coded. In total, three categories for reading programs, six categories for writing programs, one category for proprietary programs linked to paper resources, and three special needs categories were identified. Reading programs were coded as either *Drill and Practice Game*, along with subcategories of *Letters* or *General Language Skills*, or as *Talking Book* or *Interactive Talking Book*, depending on whether the program simply read to students or allowed them to interact in some way while reading.

Writing programs that helped teach writing skills were divided into *Concept Mapping* software, which provides graphical representations of ideas and allows brainstorming and *Writing Process Assistance* designed to demonstrate parts of the writing process for students. Some of these programs also supported *Drill and Practice Game* activities in some cases as well. In addition to programs which assist in teaching writing skills, several types of programs allow students to write their own work. These programs are designated as *Word Processor*, if the main function is text, or *Desktop Publishing* if the main function is text with pictures and several were designated as *Creative Writing*, because they have word processing or desktop publishing features, but also give writing prompts. Proprietary programs, designated *Program Support to Printed Materials*, and assistive technologies for special needs, such as *Text-to-Speech*, *Voice Recognition Software,* and *Word Prediction Software* were included on the list, but were not considered for my thesis.

Calculating Software Scores

Bishop and Santoro's (2005, 2006) evaluation criteria list indicators and use a five point Likert scale and were adopted (see Appendix D). For scoring the software programs and calculating the score, *Strongly Disagree* was assigned a value of 1, *Disagree* was assigned 2, *Neither Agree nor Disagree* was assigned 3, *Agree* was assigned 4 and *Strongly Agree* was assigned 5. The score for each category, *Interface Design*, *Content Criteria* and *Instructional Design*, was totalled and divided by the total possible score for each section. A score of 20% to 60% was considered not to meet expected standards, a score of 61% to 84% was considered to adequately meet expected standards, and a score of 85% to 100% was considered to meet or exceed expected standards.

Procedure and Inter-rater Reliability

Bishop and Santoro's (2006) original evaluation framework was tested by the authors with .93 average inter-rater correlation, meaning that the authors were confident that other evaluators could use their criteria and indicators and obtain similar results with the same software programs. To ensure that the evaluation criteria were applied as consistently as possible to all software used in my thesis, an independent reviewer used the criteria to test a random sample of the software. Two programs (15% of the total sample), *Ultimate Writing and Creativity Centre* and *Reader Rabbit 1*, were selected randomly and evaluated by an independent reviewer using a match – mismatch procedure. Agreements and disagreements for each criterion were tallied across categories and used to calculate total percentage agreements between the reviewers. Since Likert scales are somewhat subjective and it is difficult to match each item exactly, strongly disagree and disagree were compressed into one category, and agree and strongly agree were also compressed into one category for calculating inter-rater reliability. Any items showing significant differences in scores were discussed until consensus was reached. Using this method, inter-rater reliability was calculated to be 89%.

Bishop and Santoro (2006) presented a specific procedure with questions for evaluating software, summarized in Appendix C. The procedure provides a systematic method for accessing software in order to test all of their criteria in each category in the same manner for every software program evaluated. This procedure was followed as closely as possible with each software program that was evaluated, to ensure that each criterion and indicator was addressed in the same manner for every program. Cases where the procedure was not followed were indicated.

CHAPTER 4: FINDINGS AND DISCUSSION

Five questions guided my study of the authorized software tools for use in teaching reading and writing in the primary grades. Answers for each of the five questions are discussed in the subsequent sections. The first research question is a general question and is addressed in this introductory section. The second, third, and fourth questions are specific to each software category and frame the discussion for each subsequent section. The fifth research question encompasses all software categories and is addressed in the final section of this chapter.

Question 1: What Types of Authorized Software are Available? In total, 47 programs were identified on the compiled list of software for the provinces and territories; excluding Québec and Nunavut which both list titles in languages other than English (see Appendix B). These programs are grouped into four main categories: *Reading Programs*, *Writing Programs*, *Programs Designated For Students With Special Needs*, and *Other Programs*. Where overlap occurred in program categories, the first listed category was used. *Reading Programs* comprise 28% of the listed programs, and are divided into two subcategories: *Drill and Practice Games* (17%) and *Talking Books* (11%). *Writing Programs* comprise 51% of the listed programs, and are divided into three subcategories: *Concept Mapping Software* (15%), *Word Processing Software* (19%) and *Desktop Publishing Software* (17%). *Programs Designated For Students With Special Needs* comprised approximately 11% of programs, and *Other Programs* approximately 10%. *Other Programs* included such software as *Literacy Place for the Early Years*, which are proprietary software resources linked to books or other resources, and are not designed to be used alone, or audio discs such as *Collections 2 and 3 Audio*

Pack. Programs Designated For Students With Special Needs and *Other Programs* were omitted from this study.

Although 28% of programs were listed as *Drill and Practice Games* for reading and may be considered instructional in nature, the writing programs comprised 51% of the total and consisted of various word processing or desktop publishing programs, which are non-instructional in nature. These findings contradict those of Dwyer (2007) and Becker (1998), who stated that computers in the primary and elementary grades were being used for drill and practice games or activities, the vast majority of the software listed as approved, authorized, supplementary, or licensed software in Canada's provinces and territories does not consist of drill and practice software. Even the older programs, such as *Kid Works Deluxe* (1995), are open-ended and student-directed, rather than prescriptive drill and practice programs. In fact, the only province listing software that is purely drill and practice software for basic skills is Ontario, although some provinces list programs that incorporate some drill and practice functions, such as *Has Anyone Seen My Umbrella?* for which drill and practice games are listed as a part of the overall function. The Western Canadian Protocol for Collaboration in Education (2002), for example, lists *Has Anyone Seen My Umbrella?* and states:

> This interactive CD-ROM, based on the story of Cinderella, takes students into a magical kingdom where they can view a 10-minute animated movie, read a 40-page storybook (*Has Anybody Seen My Umbrella?* by M. Ferguson), write their own tales, listen to theme-related music and instruments, and explore words through games (p. 76)

which indicates that drill and practice word games comprise only part of this software program.

The listings for software in each provincial or territorial jurisdiction comprise a base listing of what is considered useful or appropriate. Schools make their own decisions as to which software to install and use on their computers. For example, *Reader Rabbit 1*, though listed as licensed software in Ontario, might be used in schools in Alberta even though it is not listed as an authorized software resource for Alberta. Conversely, not all software that is listed may be used in all schools within the jurisdiction, however, it would be difficult to ascertain, with complete accuracy, which software programs are being used in schools in Canada. Some provinces, most notably New Brunswick and Newfoundland and Labrador, no longer maintain central software listings, and each district evaluates its own software. It may be, then, that schools in Canada are using drill and practice software, such as Dwyer (2007) found in her study of Australian teachers' attitudes towards the use of technology in their classrooms, but my study is confined to the authorized software listings.

In the subsequent sections, a small selection of 13 of the programs that are listed throughout the Canadian provinces and territories is discussed in light of the second, third, and fourth research questions. These programs were chosen based on availability, as many programs were for order, however in most cases they represent typical programs for each subcategory of reading and writing software. Reading programs are presented first, followed in a subsequent section by writing programs. In each section, the second, third, and fourth research questions are considered and frame the discussion of results.

Software Claiming to Teach Reading or Facilitate Reading Development

Two major types of reading software emerged from the software programs assessed, and they have been organized, for the purposes of discussion, into two groups: *Drill and Practice Games*, and *Talking Books*. As stated in the previous section, at times the distinction between the various types of reading software is vague since one software program might have multiple functions. Where programs overlapped, *Talking Books* was given preference over the *Drill and Practice Games* category which was reserved for pure drill and practice games, because in these cases, generally the drill and practice portion of the software supports the book being read.

Drill and Practice Games

Several older programs for teaching reading skills rely on drill and practice games to teach basic reading skills such as letter recognition or letter-sound correspondence. Each of the reviewed programs is listed as a resource for Ontario, and no other provinces or territories list pure drill and practice programs on their authorized software lists. (For convenience, all software program titles are abbreviated with acronyms in the *Findings and Discussion* chapter. An alphabetical list of program title acronyms is presented in Appendix E).

Question 2: What do the manufacturers claim their products do and do the products support the claims? The software programs reviewed for the *Drill and Practice Games* category have several similar claims. *A to Zap!* (*AtZ*), *Bailey's Book House* (*BBH*), and *Reader Rabbit 1* (*RR1*) are designed for preschool children and students in kindergarten and first grade, so the manufacturers' main claims to improve letter and word recognition, particularly using picture and sound referents, are directly addressed by the programs' activities and are appropriate because children typically are taught these skills in these grades. In addition to letter and word recognition skills, the manufacturers list supplementary skills in their educational claims. Matching lower and uppercase letters, knowing alphabetical order, understanding that letters combine to make words, recognizing the beginning sounds of words, and exploring the comparative and superlative forms of adjectives are additional claims made by *AtZ*. Increased reading fluency, spelling pattern recognition, visual discrimination, and critical thinking are additional claims made by *RR1*. *BBH* lists no specific educational claims, but does address rhyming words, beginning letter sounds, descriptive words, colours, and prepositions with activities. In each program, the supplementary skills are not addressed directly in most cases, and when they are specifically addressed through activities, letter and word recognition are given precedence.

Table 1 presents the results of assessing three *Drill and Practice Game* titles using categories and indicators adapted from Bishop and Santoro's (2005) *Early Reading Software Evaluation Form* (see Appendix D). The percentage values were obtained using the procedure established in the Methodology chapter, by adding the scores for each indicator in each evaluation category (*Interface Design*, *Content*, and *Instructional Design*) and dividing by the total possible score for each. Each software category presented in subsequent sections was assessed in the same manner. As 20% is the minimum possible score in each category, a score of 20% to 60% was considered not to meet expected standards, a score of 61% to 84% was considered to adequately meet expected standards, and a score of 85% to 100% was considered to meet or exceed expected standards.

Table 1

Percentage of Total Possible Score by Evaluation Category for Drill and Practice Games

Game Titles	Interface Design	Content	Instructional Design
A to Zap!	67	64	51
Bailey's Book House	77	86	71
Reader Rabbit 1	88	73	69

As can be seen from Table 1, scores for each program show significant range in *Interface Design* (21 percentage points), *Content* (24 percentage points), and *Instructional Design* (18 percentage points). All programs meet expectation for *Interface Design*, however the lowest score (67 percentage points for *AtZ*) barely meets expectation and the highest score (88 percentage points for *RR1*) exceeds expectation. *Content* scores follow a similar pattern. The lowest score (64 percentage points for *AtZ*) barely meets expectation and the highest score (86 percentage points for *BBH*) exceeds expectation. *Instructional Design* scores were lowest for each program of the three evaluation categories. One program did not meet expectation for *Instructional Design* (*AtZ*), and no program exceeded expectation. Overall, *Drill and Practice Games* scored slightly higher on *Interface Design* than *Content*, and much lower for *Instructional Design* in every instance, suggesting that interface design is the most important consideration for manufacturers, followed closely by practicing skills. Each of these programs is designed for home and school use, so *Instructional Design* may not be as important to game designers as having a pleasing interface and addressing content skills because manufacturers market their products as entertaining and educational games and not as instructional tools designed for classroom use.

The subsequent discussion outlines the major features influencing the scores obtained for the reviewed software programs. Although they cover some similar content, the effectiveness of programs in teaching the skills which they address varies, causing their *Content* scores to differ greatly. These programs are designed to function differently, accounting for the varied scores across categories *Interface Design* and *Instructional Design* categories.

The highest scoring program for *Interface Design* was *RR1*, which uses the metaphor of a word factory where students play four phonics games. The screen layout is simple, with all program functions being handled through simple menus incorporating text and pictures. The colours are bright, although the use of patterned graphics on the page could distract students. Kindergarten and first grade students would likely feel comfortable and confident in using *RR1*'s interface independently to play the game due to the consistent, simple screen layout, the availability of auditory directions and help functions, and the high level of interactivity. *BBH* also scores adequately on *Interface Design*, uses the metaphor of a house, and offers two modes of difficulty for students to either practice or test their skills. In the default practice mode, each response presented is correct and students may choose from among any of them, and in the challenge mode, only one response is correct for each question. Students change between modes by clicking on the picture frame in each activity. Children who can read and those who are not yet reading could enjoy and feel challenged by the activities, and navigate easily through the activities in either mode, due to the consistent layouts with computer-generated voice prompts. *AtZ* uses the metaphor of alphabet blocks for buttons and each block either allows the interface to be modified or links to a game or other activity. Some activities have associations to the letters represented such as *M* for *music*, which plays music but some activities, such as H for *hammer*, which is a spelling activity, are not closely associated with their letters. Children using the program for the first time must guess the functions or wait for the animation to play to ascertain the function of the letter, that is, which activity is associated with each alphabet block button. There are twenty-six activities in this program, however neither instructions nor help in choosing activities is provided, consequently lack of assistance was a major factor in *AtZ's* lower *Interface Design* score.

In the *Content* evaluation category, the highest scoring program was *BBH.* The activities give ample practice of letter recognition, letter-sound correspondence, beginning sounds, and rhyming skills in a non-restricted fashion in the default mode and permit testing of students' knowledge in the challenge mode. For example, in the *typewriter* activity, students type letters and hear their sounds, and in the challenge mode, they are told letters and asked to find the matching printed letter. Students may choose the level of difficulty and move between modes without sacrificing program functionality. RR1 also scored adequately on *Content.* As a phonics-based program, *RR1*'s games directly practice letter and sound matching skills using consonant-vowel-consonant (cvc) words. One strength is that the levels of difficulty are adjustable by students to fit their abilities. For example, less capable students might match a picture of "cat" to the word *cat*, and more capable students might match the picture of "cat" to the middle letter, *a*. Students choose how many letters are present in the words they match, so that both less and more advanced learners are challenged without changing how the individual games function. According to *AtZ*'s manufacturer, "...these explorations integrate different curriculum areas by focusing children's attention on their own interests through game-playing and problem-solving goals rather than on repetitive drill and practice" (Creative Media Applications, 2005, p. 6), yet many activities ask students to perform basic skills such as matching letters or spelling simple words. Although there is no predetermined order for performing the activities, they are by no means open-ended activities where students control all program functions. In the H *hammer* activity, for example, the letters for each word must be entered in the correct order, and if the program expects the word *music* for *M*, it will not recognize other words beginning with M, like *moo*. With 26 different activities available covering a wide variety of topics including reading, spelling, letter-sound correspondence, mathematics, and music, none of these topics receives more than superficial treatment within the program, with the exception of letter-sound correspondence and beginning sounds, thus contributing to *AtZ*'s low *Content* score.

Each of the programs in the *Drill and Practice Games* category scored lowest on *Instructional Design*. The highest scoring program for *Instructional Design* was *BBH*, although *RR1* had a similar score. The main strengths of these programs are the computer-generated voice and how student errors are handled. Activities are introduced by computer-generated voice and many instructions are also given auditorily. When students make errors, the prompts are repeated aloud, and students are given further chances to practice the skill. In one activity in *BBH*, for example, students differentiate between *over*, *on*, *behind*, and other prepositions. If they choose incorrectly, the incorrect response is removed and students may repeat the activity until the correct answer is selected. In contrast, *AtZ*, which claims to be exploratory and allow students choice while learning, only superficially covers most topics and also has students sit passively during many activities. The program offers limited help to students when required, and does not indicate a difficulty level for the activities.

The *Drill and Practice Games* contain many of the same flaws which contribute to the low scores for *Instructional Design*. Although introduced by voice in most cases, the importance and relevance of the activities are not explained to students. *BBH* and *RR1* have various difficulty levels, yet neither they nor *AtZ* track student progress through the activities in order to adjust the difficulty levels if the selected level or the default level is inappropriate for student ability. For example, if the students begin at the default difficulty level for the *Sorter* activity in *RR1*, they match beginning sounds in words to the beginning sound of the example word presented by the program. Once the student has progressed through several rounds, the program does not advance to the second difficulty level where the student sorts words according to ending sound. Students must manually select the difficulty level for each activity and are not prompted by the program to adjust the difficulty level if many errors are made, for example.

In addition to tracking student progress, educational programs should provide feedback to students and teachers (Bishop & Santoro, 2006), to enable future educational goals or activities to be planned, yet none of these programs offer the function. The flaws presented are inherent in the game genre because the game philosophy is different from the instructional philosophy of many classrooms, especially when the game is designed for home and school use, yet these flaws detract from the software programs' instructional value in a classroom.

Question 3: Are reading and writing prerequisite skills for using the software programs? The children for which these *Drill and Practice Games* are designed range from preschool age to first grade. As most of the programs' functions are introduced with voice, students are not required to read directions and question prompts independently. In addition, especially with *BBH* and *RR1* where incorrect responses are simply ignored and eliminated from the game, students could click any response until the correct response is selected which amounts to nothing more than gradual elimination merely by clicking. Although *AtZ*, *BBH, and RR1* are instructional in purpose or at the very least provide opportunity for students to practice their phonics skills, the inability to practice skills in a natural context could hinder transfer of skills outside the game context and into students' learning and may mean that students are not retaining the skills which the programs are designed to teach.

Question 4: Are these software programs appropriate and useful to supplement classroom reading and writing instruction? The reviewed programs for *Drill and Practice Games* address skills that are appropriate to the grade levels of the students for which they are intended, specifically letter and word recognition of simple words. Although students practice the skills, there is no evaluative feedback and no tracking function for student performance, so their usefulness as instructional programs is limited. A specific flaw of these games is that students do not have to read to play the game, because the programs often eliminate incorrect responses until the correct response remains, and do not automatically perform error analysis to modify program function to suit student needs. Another flaw of these games is that they have set functions that are pre-programmed and limit the ability to replay the game with new experiences or new words. *RR1* cannot be adapted to use vocabulary words from classroom instruction instead of the predetermined cvc words. The rhymes and stories created by *BBH* cannot be adapted or changed outside of specific parameters to allow more characters or different rhymes. Teachers using *AtZ* may input words for the spelling function, but these words are not supported by computer-generated speech.

Talking Books Software

Approximately 11% of the software programs were designated *Talking Books Software*. These programs typically have two main parts. First, the program contains an electronic version of a storybook which becomes the theme for the program. Students read the story along with the computer-generated voice and often the computer highlights words as they are spoken. Activities such as drill and practice games, writing activities, or songs are also included and are linked to the story's theme. Students using *Talking Books Software* often may choose between having the story read completely by the program or reading the story independently (Littleton, Wood, & Chera, 2006).

Although an attempt was made to find the titles listed as *Talking Books Software*, no copies were available for purchase. *The Cat Came Back* and *Sitting on the Farm* were only listed only on sites in the United Kingdom with very limited availability and no guarantee of overseas shipping in reasonable time, The National Film Board has stopped producing *Has Anybody Seen My Umbrella?*, and *Chicka Chicka Boom Boom* was back ordered. *Changes: The Turtle's Teachings* was procured, and although it is listed as a CD-ROM, it is simply an audio retelling of the story on compact disc. The story is not displayed onscreen and no activities are included to support the theme of the story.

Software Programs Claiming to Facilitate Writing Development

Many of the authorized software programs for teaching writing in Canada actually facilitate or scaffold the writing process, and do not directly teach writing skills. These software programs are designed to make the writing process easier or more enjoyable for students. Three major types of writing software emerged from the programs assessed, and they are organized, for the purposes of discussion into three groups: *Organization of Ideas (Concept Mapping Software)*, *Word Processing Software*, and *Desktop Publishing Software*. The final two categories have significant overlap of function, but there are also significant differences. *Word Processing Software* programs typically provide more sophisticated text-editing functions with limited graphics or layout options, and *Desktop Publishing Software* programs generally provide simple word processing capabilities with more sophisticated graphic and text layout features.

Organization of Ideas (Concept Mapping Software)

Several of the programs reviewed are designed to help students better organize their ideas and clarify their thinking on topics so that they communicate more effectively with their audiences. These programs have been labelled *Concept Mapping Software*, as their main focus is to create concept maps and allow students to transfer their concept maps, or webs, into an outline for organizing their writing.

Question 2: What do the manufacturers claim their products do and do the products support the claims? There is much similarity in function between the *Concept Mapping Software* titles reviewed for this thesis. *Kidspiration 2* (*K2*), *Inspiration 8* (*I8*), and *SMART Ideas 5.0* (*SI5*) are very similar programs both in interface design and the way concept maps are used to create outlines, and *Draft: Builder Solo* (*DBS*) has many similar functions, although the program works in reverse and outlines are used to create concept maps. Consequently, the manufacturers' educational claims about their programs are also very similar. The two main claims put forth by *Concept Mapping Software* manufacturers are the improvement of writing skills and thinking skills. *K2*, *I8*, and *SI5* claim to strengthen organization of ideas and increase vocabulary when writing. For thinking skills, they claim to clarify thinking, enhance understanding, promote visual learning, and increase retention of concepts. *SI5* also claims that using the software allows students with weak conventional writing skills to communicate more clearly. In addition to writing and thinking skills, *K2* lists reading comprehension and word recognition skills, although these skills are secondary to the writing skills. *DBS* uses language closely related to the writing cycle to describe the program's educational functions. The manufacturer claims that this program is useful for teaching organization and prewriting strategies, how to draft and revise by adding descriptive language and elaborating on main idea, and how to edit and publish by editing spelling at a developmentally appropriate level. In addition, it is claimed that the program teaches students to use paragraph format, to arrange ideas in sequential order, and to write coherently. For reading, *DBS*'s manufacturer claims that the use of the program helps students develop a purpose for reading.

Unlike the *Drill and Practice Games*, *Concept Mapping Software* does not directly instruct students in skills. Unless teachers create templates for students to use, the programs are completely open-ended with unlimited opportunity for students to organize ideas within the confines of program function. Most of the claims are adequately supported by how the programs function, with the exception of reading claims, which appear only indirectly supported.

In Table 2, the results for the four *Concept Mapping Software* programs using the adapted categories and indicators from Bishop and Santoro's (2005) *Early Reading Software Evaluation Form* are presented. Percentage scores in each evaluation category (*Interface Design*, *Content*, and *Instructional Design*) were determined as previously discussed and the standards for determining whether programs meet or exceed expectation are consistent with those from Table 1.

Table 2

Percentage of Total Possible Score by Evaluation Category for Concept Mapping Software

Software Titles	Interface Design	Content	Instructional Design
Draft: Builder Solo	81	76	89
Inspiration 8	75	87	60
Kidspiration 2	80	80	61
SMART Ideas 5.0	74	80	59

As shown in Table 2, there is much less range on *Interface Design* (seven percentage points) and *Content* (11 percentage points) than *Instructional Design* scores where the range of percentage points is 30 between the lowest and highest programs. All programs meet expectations for *Interface Design* with scores ranging from 74 percentage points (*SI5*) to 81 percentage points (*DBS*) and *Content* with scores ranging from 76 percentage points (*DBS*) to 87 percentage points (*I8*), and *I8* actually exceeds expectation for meeting the program's listed educational goals. Three programs either do not meet expectations or only just meet expectations for *Instructional Design* with scores ranging from 58 percentage points to 61 percentage points, but the fourth, *DBS*, actually exceeds expectations for this evaluation category with a score of 89 percentage points. A more specific discussion of the features influencing the scores for each program in each evaluation category follows.

Three of the reviewed programs, *K2*, *I8*, and *SI5*, function very similarly and their interfaces are designed to look alike as well. As they are so similar in function, interface design, and content, the programs score similarly on all evaluation categories. Program strength appears to be in *Interface Design* and *Content*, and being much weaker in *Instructional Design*. The differences between program scores for each category are due to the programs being directed towards different age groups, with *K2* being designed to be most appropriate for primary students. *DBS* scores similarly to the other programs for interface design, is slightly weaker in *Content* because of numerous and specific claims about educational function, but scores highest on *Instructional Design* as it is designed specifically for school use and incorporates features to track student progress across tasks.

Each of the programs in this software category scored high on *Interface Design*. *K2*, *I8*, and *SI5* all have adequately designed interfaces for their intended audiences, and use consistent layouts that are generally fixed and unable to be modified (like button size). *K2* incorporates computer-generated voice and pictures into all functions and directions, which increased scores on *Interface Design*, and deals with topics more suited to younger audiences, such as fairytale creatures. Primary students would be able to use *I8* and *SI5*, but their graphics, menu functions, and lack of voice support for button functions make these programs less appealing for primary students. *DBS* also uses a simple layout. Students may enter text, but not pictures, sounds, and animations, and the tools are more suited to text-based functions. *DBS* does not permit graphics or sound effects to be incorporated into concept maps, for example, but *K2* treats graphics and sounds in the same way it treats textual information. Of the four programs, the interface for *K2* is easiest to navigate and most appealing for primary students, but the ability to modify voice feedback, button size, and other interface resulted in higher scores for *DBS.*

Each program scored adequately on *Content*, based on the two main skill areas of thinking skills and writing skills that were identified in their literature as the main skills addressed by the software. Students use writing and thinking skills in creating the concept maps, rather than being taught these skills by the programs. Students organize ideas and forge conceptual links between them by using interconnected symbols, text, pictures, and sounds to present their ideas visually. In each web, students begin with their main topic and continue by adding symbols containing text, choosing pictures from the libraries to represent ideas, and incorporating sound clips from within the program to add details describing their main topic. In most cases, students may incorporate their own pictures or record their own voices and sound effects for their concept maps. As students add more symbols, text, pictures, and sounds to each idea branching from the main topic, they add more detail. Ideas are linked with arrows or with pictures. For example, if a student created a character map for a story, the main idea might be the main character, which connects to pictures representing the main character's friends. The connecting arrow might read "goes to school with". After students create their concept maps, the programs convert them into the linear format of an outline, so that students can use their concept maps to plan their writing. *DBS* works in reverse to the other *Concept Mapping Software* programs, because students begin with the outline and the program converts the information into a concept map that uses words only, not pictures or sounds. Unlike the other programs, *DBS* also has limited word processing features included for beginning to draft ideas into paragraph form.

The *Content* skills listed but not used directly in creating maps were reading skills. *K2* lists word recognition and comprehension skills as addressed skills but the program mainly targets writing skills and not reading skills when creating concept maps. Word recognition and comprehension might be said to be addressed by the computer-generated voice reading the labels of the pictures, and by the text-to-speech functions that read back student writing but the claim that this program enhances word recognition and reading comprehension is somewhat tenuous. The text-to-speech function is limited because it reads words phonetically, does not read words with natural voice or intonation, is somewhat choppy, and would serve as a poor model of reading fluency. For example, each sound is pronounced in a word with equal emphasis, rather than how it sounds in normal speech, such as "prin-suss" instead of "prīn-cess" with the proper stress on each syllable. Unlike *DBS*, which allows students and teachers to modify the voice slightly to correct pitch, speed, or mispronunciation, *K2* offers no capability to correct mispronunciations, change reading speed, or improve the quality of the text-to-speech function. *DBS*'s claim to help students develop a purpose for reading seems even more tenuous than *K2*'s claim to improve word recognition and comprehension unless, for instance, teachers create templates for reading response activities such as having students create an outline of a character sketch for a story they have read.

DBS, *I8*, *K2*, and *SI5* do not directly teach writing skills and appear only to facilitate the development of organization and writing skills. Since they are not designed to be directly instructional, these programs score low on the *Instructional Design* category, with the exception of *DBS*. Bishop and Santoro (2006) noted that the ability to track and evaluate student growth and provide directed feedback was important for educational programs to be considered instructionally sound, yet the nature of *Concept Mapping Software* makes these functions very difficult to achieve. Instead, teachers evaluate the product or observe and evaluate the process used and since the help files are not context or content sensitive, no feedback to students is provided. Furthermore, the open-ended nature of webbing does not easily allow computerized assessment or tracking based on pre-specified criteria. Although *I8*, *K2*, and *SI5* do not track student progress, *DBS* tracks word count, average word length, sentence count, average number of words per sentence, and use of low frequency words in the integrated word processing feature and reports them in chart or graph form. The program warns teachers that these graphs and charts give a synopsis of writing growth in the selected documents, and should not be used to assign grades, however this unique tool would be quite useful for teachers to monitor student progress over multiple tasks. *DBS* gives teachers the option to manage assignments and track student progress from a central account which accesses student work, allows options to be changed, and permits teachers to offer feedback directly on student work.

Question 3: Are reading and writing prerequisite skills for using the software programs? Writing is prerequisite for using all of the programs designated as *Concept Mapping Software*. Although students might create concept maps containing only pictures and sounds without text, the outline function and the templates encourage students to write and expand their ideas. *SI5* specifically claims to help students who have limited writing ability to communicate more effectively and it is the case that using concept maps requires less writing, especially if pictures and sound are incorporated, but after helping design the outline for writing, students must write the sentences independently to convey meaning.

Students could operate *K2*, *I8*, and *SI8* without reading, but typically students will read directions, button names, and menu options even if where they are also represented by pictures or described by computer-generated voice. *DBS* is text-based and all assignment directions are textual so reading is required for using the program. In addition to reading directions or buttons, students typically read their own work as well.

Question 4: Are these software programs appropriate and useful to supplement classroom reading and writing instruction? For a primary classroom, the most suitable *Concept Mapping Software* program for student interest and ease of use from among the four programs analysed is *K2*. The most suitable program as an instructional tool is *DBS*, however the lack of graphics would be less appealing to primary students, and unless the teacher establishes the outline format in advance, the program would be more difficult to use than the other three. It is reasonable to conclude after examining interface design, graphics, and program functions that *I8* and *SI5* appear designed for older students or adults to use. Although *I8* appears on the general software lists for Nova Scotia, Prince Edward Island and Yukon Territory, and *SI5* appears on the list for Ontario (See Appendix A), these provinces and territory do not distinguish between grades, in many cases, when listing software that is authorized for kindergarten through sixth or seventh grade.

The non-instructional nature of *Concept Mapping Software* indicates that these programs should be considered tools for making the writing and idea-generation process easier rather than instructing students in the writing process or in how to generate ideas for writing. Students learn to organize their ideas by creating concept maps, but there is no correct or incorrect way to organize ideas. Students choose topics and without requirements for the type of information to add unless the teacher has specified that certain types of information be included. Since *K2*, *I8*, and *SI5* begin with blank webs, the products that result are limited only by the users' ideas. One drawback is when students do not have ideas to put into the diagram or the outline view. In this instance, the software would offer no help without further research being done by the student. The software offers support on how to create webs, but does not impose formatting or layout ideas on students' own work. Unrestricted formatting options help students who are adept at creating their own webs and outlines, but hinder those who require more assistance with getting started or organizing their ideas. In contrast, *DBS* is designed so that teachers can add locked text (text that may not be modified by the student, but does not show on the printed page) to give instructions, suggest ideas, or provide feedback for areas to change which makes *DBS* better than the other *Concept Mapping Software* for instructional purposes when teaching writing. These programs do not address reading unless students read their own or others' writing or program directions and no reading instruction functions are provided.

Each of the *Concept Mapping Software* programs comes with pre-loaded templates to be used by teachers as starting points for organizing common functions or activities. There are several categories to choose from, and the templates for *Reading and Writing Activities* in *K2*, for instance, include typical activities students in the primary grades might be asked to perform. For example, in the *Adjectives* template, students are challenged to provide adjectives to describe their chosen topic and then to write sentences for each adjective in the outline view. The program prompts students to think of words that use the five senses, such as *green, rustling* leaves, *rough* bark, *fresh* smells, and *sweet* apples to describe a tree. In this program, the category *Reading and Writing Activities* appears to be misnamed, as the primary focus is on writing skills or writing about what has been read, but virtually nothing is done to address reading, with the exception of having students read their own writing or having students read books or stories and then complete concept maps using the software. The templates in *K2* and the other *Concept Mapping Software* are very general, focussing on giving directions for how to use the template and offering tips for writing ideas, but not instructing students on how to perform the skills. For instance, for *Parts of Speech* in *K2*, students classify vocabulary words as nouns, verbs, adjectives, or adverbs, yet these categories are not explained, and no corrective feedback is provided if students make errors. The directions encourage students to write five sentences using the vocabulary words with no feedback to indicate whether the task was performed correctly. The lack of corrective feedback and specific skills instruction is, perhaps, the greatest reason why *Concept Mapping Software* can be said to facilitate writing for those who are already proficient writers, or to help those who need assistance with idea generation and organization, but not to teach writing skills.

Less proficient writers would require further assistance to write sentences that incorporate the ideas from the outlines and meaningfully communicate ideas.

Word Processing Software

The programs reviewed in this section are designed specifically for children to use at home or school. Unlike traditional word processing programs, these programs are "talking" word processors, that is the programs read students' text either upon request, for *Clicker 5* (*C5*), or automatically by letter, word, sentence, or paragraph for *Write: Outloud Solo* (*WOS*). *C5* has desktop publishing functions and students may use the cells to place text and pictures in different layouts on the screen, however the manufacturers identify *C5* as a word processing program and not a desktop publishing program. The main function of *C5* is for the Clicker writer function to be used to input text into a word processing document, thus it is included as a *Word Processing Software* program.

Question 2: What do the manufacturers claim their products do and do the products support the claims? Of the two word processors that were reviewed, *WOS* more explicitly states the manufacturer's claims for educational benefits. *WOS*'s manufacturer claims that the program teaches students to write in paragraph form using indentation, topic sentences, main idea, introductions, and conclusions. In addition, students edit grammar and spelling at developmentally appropriate levels and use reading skills and strategies to understand a variety of informational texts, according to the manufacturer. The manual for *C5* lists no specific educational benefits, so benefits claimed by other manufacturers of similar programs such as word processing software or desktop publishing software were adopted. The main claims that were adopted include: students write in full sentences, edit for syntax and grammar, and improve vocabulary development. In keeping with other programs making significant use of text-to-speech functions, such as *K2*, the claim that the *C5* talking word processor helps improve reading fluency was also adopted.

The results of assessing these two *Word Processing Software* titles are presented in Table 3. *Interface Design*, *Content*, and *Instructional Design* values are presented as percentages of total possible scores which have been calculated in the same manner as previous tables, and the benchmarks for meeting and exceeding expectations are also consistent.

Table 3

Percentage of Total Possible Score by Evaluation Category for Word Processing Software

Software Titles	Interface Design	Content	Instructional Design
Clicker 5	82	68	72
Write: Outloud Solo	81	78	89

As evident in Table 3, both of the reviewed talking word processors scored adequately on *Interface Design*. The scores for this evaluation category varied by only one percentage point. *Content* and *Instructional Design* scores varied more dramatically with ranges of 10 and 17 percentage points respectively. Both programs met expectations on *Content* and *Instructional Design*, and WOS exceeded *Instructional Design* expectations. The features influencing the scores for both programs are addressed next.

Both *C5* and *WOS* score at the upper limit of adequate for *Interface Design*. These programs were designed specifically for children so functions are easily found, buttons use graphics rather than words, and features, such as the spell-checking, are designed to be useful for students. With *C5*, students input words by clicking on cells (boxes) that contain words or pictures, using a word bank that contains an alphabetical list of words, typing words in with a keyboard, or clicking the mouse on a virtual keyboard. As it is also designed for persons with special needs, *C5* has a highly adaptable interface, where menu items, buttons, and toolbars may be resized, the amount of support available may be changed for functions like the spellchecker, and the text-to-speech settings may be modified. *WOS*'s interface is not as adaptable as *C5*'s interface, however the amount of support such as auditory and visual cues for misspelled words and the text-to-speech settings may be modified.

In considering *Content*, neither *C5* nor *WOS* directly teach the skills they claim to facilitate, however the skills are used to varying extent when creating text. *C5* does not indicate educational objectives, so objectives from similar programs were adopted for *Content*. *C5* scored adequately for helping vocabulary, spelling, grammar, and syntax development, especially through the use of pictures and word labels. Since most talking word processing programs or text-to-speech enabled desktop publishing programs also list reading fluency as a goal, reading fluency was included as an evaluation criterion for *C5*. Although students may read along with the computer-generated speech or record themselves reading text, reading fluency is not directly addressed by the software as a primary skill. *WOS* is more specific about educational objectives, listing paragraph writing, editing and publishing, and reading skills and strategies as the three major areas of focus. *WOS* can be linked to *DBS* and students may incorporate webbing and outlines into their work. Since both *DBS* and *WOS* have functions which teachers may use to establish the structure for writing through locked text, the paragraph format may be directly addressed with the use of *WOS*, but only if the teacher creates a template. The paragraph format is not automatically addressed without the use of templates. Editing and publishing are assisted by visual and auditory cues for misspelled words, links to a dictionary and a phonetic homonym identification tool, the capability to modify colour, size, or font face, and the inclusion of pictures.

Both *C5* and *WOS* meet or exceed expectations on *Intructional Design* because they include functions which can be used to directly teach writing skills such as spelling, organization of ideas, or paragraph structure, although these functions are not automatic. *WOS* scores higher than *C5* on *Instructional Design.* The main reason for the discrepancy on scores for this evaluation category is that although *C5* is more adaptable to student needs, *WOS* tracks student progress by recording word and sentence use, and teachers may edit or provide comments directly on student work from a central account. These tracking, editing, and commenting functions are ideal for teachers to use for monitoring student progress and providing directed feedback.

Question 3: Are reading and writing prerequisite skills for using the software programs? The function of these word processing programs is to facilitate writing, so writing is required as prerequisite for using the programs. *C5* is unique among all of the reviewed programs in every category, because students have a choice of how to input text. For example, students may use a physical or virtual keyboard, or they may click words from the word bank or words and pictures from the cells. Regardless of the input method used, the words and pictures selected are inserted into the word processing interface. The word bank and cells would be particularly useful for teaching vocabulary words from a story or words that students commonly misspell, by having students incorporate the target words into sentences they type with the keyboard. *WOS* only provides keyboard input, although it will link to concept maps created with *DBS* so students can plan their writing in advance. Students using the text-to-speech function can go back and listen to their writing to make sure that their writing makes sense and flows, thereby improving overall writing quality, however the text-to-speech function hinders the ability to type quickly in *WOS*.

Reading is not a prerequisite skill for either word processing program reviewed for this software category, however both programs will read back student writing, which may indirectly help students improve their reading fluency as they read along with the program while it highlights the words. Students using *C5* can import their own sounds, so they might record their own voices in another program, and then import them into the document to improve reading fluency by practicing rereading their own text.

Question 4: Are these software programs appropriate and useful to supplement classroom reading and writing instruction? Both of the reviewed talking word processors are extremely versatile tools and appropriate for use in the classroom. Although neither is directly instructional and their interfaces might not be as appealing to students as the *Desktop Publishing Software* discussed in the next section, students would be able to use either program to produce written text. *C5* is designed for children of any school age, or students with special needs, and *WOS*, while not explicit about age or grade level, does suggest by its design and adaptability, that it could be used with any school-aged students.

An important benefit of using these programs for teachers is the option to format student assignments through the use of templates in *WOS* and pre-determined text in the cells for *C5*. In *C5*, the teacher could enter vocabulary words into the cells for students to use in their writing or use the cells to have students match pictures to vocabulary words. The teacher could also import a diagram and have students type or select labels for each cell. *WOS*, especially when used in conjunction with *DBS*, gives teachers the option to structure outlines, webs, and word processing documents to fit assignment expectations. Although *WOS* is text-based, students may add pictures into the written product. Students using *C5* may incorporate pictures, sounds, and animations, making it much more multimedia-based than *WOS*.

Both *C5* and *WOS* are newer programs, released in 2005 and 2006 respectively, and take advantage of the multimedia functions of computers. *C5* and *WOS* both have tools specifically designed for students' needs in order to help them modify text without using overly complicated functions. For example, each program has spell-checking capabilities. Both can be set to underline misspelled words to cue students, and *WOS* plays an auditory cue if words are misspelled, has a built-in dictionary, and a phonetic homonym identification tool to help students select the proper word by choosing between similar-sounding words, such as *there*, *their*, and *they're*. The spellcheckers and other program functions are designed specifically for student needs such as the homonym identification tool in *WOS* or the word bank function of *C5,* which help students find and use words when they might not otherwise know which words to use in their writing.

Desktop Publishing Software

In *Desktop Publishing Software* designed for adults or even for older students, the emphasis of the software programs is generally on providing templates for various layouts such as pamphlets, posters, articles, cards, and other products where the user wishes to arrange text and graphics to create a visual effect. *Desktop Publishing Software* programs for students in the primary grades often use the context of storybooks, although other contexts are available, such as the different "lands" visited in *The Ultimate Writing and Creativity Centre*.

Question 2: What do the manufacturers claim their products do and do the products support the claims? Programs selected for this software category are very similar to *Word Processing Software* on function and literacy skills, except that *Desktop Publishing Software* programs use a familiar theme, such as storybooks, to present program functions in a user-friendly manner. The claims made by the software programs' manufacturers are also similar. A common claim made by the manufacturers of software programs reviewed as *Desktop Publishing Software* is that through using these programs, students demonstrate and use their understanding of the writing process (pre-writing, writing, revising, editing, and publishing). In practice, most of these programs typically focus on writing and publishing, ignore pre-writing and revision, and perform editing functions for students rather than helping students to edit for grammar and spelling independently. Helping students to generate ideas for pre-writing, promoting storytelling skills, and encouraging collaborative writing are also claims made by these programs. One program even claims to assist student development of reading and writing skills through the language-experience approach, where students' writing becomes the material used for reading instruction.

Four software programs were reviewed for this software category. *Easy Book Deluxe* (*EBS*), *Kid Works Deluxe* (*KWD*), and *Storybook Weaver Deluxe* (*SWD*) use storybook themes and interfaces, but *The Ultimate Writing and Creativity Centre* (*UWCC*) has students assume the role of explorers who investigate different lands. The final table, Table 4, presents the results of assessing the four *Desktop Publishing Software* titles using the adapted form of Bishop and Santoro's rubric (2005). Each software program was assessed in a manner consistent with the previous tables for the evaluation categories of *Interface Design*, *Content*, and *Instructional Design*, and the standards for determining whether a program met or exceeded expectation were also maintained for this software category.

Table 4

Percentage of Total Possible Score by Evaluation Category for Desktop Publishing Software

Software Titles	Interface Design	Content	Instructional Design
Easy Book Deluxe	68	65	47
Kid Works Deluxe	83	73	54
Storybook Weaver Deluxe	72	70	46
The Ultimate Writing and Creativity Centre	71	83	75

As can be seen from Table 4, the percentage scores for each program show great variability across all evaluation categories. Although each of the programs met expectations for *Interface Design* and *Content*, three programs did not meet expectations for *Instructional Design*. Scores for *Interface Design* range by 15 percentage points, from the lowest program score (68 percentage points for *EBD*) to the highest (83 percentage points for *KWD*). *Content* scores range by 18 percentage points, from the lowest program score (65 percentage points for *EBD*) to the highest (83 percentage points for *UWCC*). Scores on *Instructional Design* showed the greatest range (29 percentage points), from the lowest program score (46 percentage points for *SWD*) to the highest (75 percentage points for *UWCC*). Even though three of the programs use the same theme and program design, each has specific features that influenced scores for each of the evaluation categories. The factors influencing all four programs' scores are discussed next.

The highest scoring program on *Interface Design* was *KWD*, one of the storybook programs. *KWD*, like *SWD* and *EBD*, casts students in the role of storybook authors who create stories to share with audiences. The main screen layout in each program is a storybook page, although *SWD* and *EBD* mix text and graphics on each page but students using *KWD* choose between full pages of text or graphics. All programs utilize picture icons for commonly used functions, and text-to-speech functions are available to read stories to students, although the activities, instructions, and buttons are neither announced nor explained with computer-generated speech. Minor differences in interface contributed to KWD scoring highest on *Interface Design*. For example, menu options for both *EBD* and *SWD* software programs are textual with no pictures, but *KWD* combines text and pictures. The help function in *KWD* explains the buttons and functions aloud if students click on it. In addition, students using *KWD* have more options for editing story graphics than provided by the other two storybook programs. *SWD* also scored adequately on *Interface Design*. The program graphics are crisp, the sound features have satisfactory quality, and students have the option to make limited choices for text and picture layout. In contrast to *SWD*, however, *EBD* has far more text and graphics layout options, such as having the picture in a column to the left, and the text in a column to the right, but the program graphics and the interface graphics are extremely grainy and pixelated, contributing to *EBD*'s low score on *Interface Design*.

When using *UWCC*, which does not follow the storybook motif, students travel to four different lands, including the rainforest, the desert, space, and the ocean. The program offers information and writing ideas for each land as well as help and ideas for each stage of the writing process from pre-writing, through drafting, revising, editing, and publishing. The main layout is a blank page, onto which students may add text, pictures, or notes. A notebook function is also provided where students can plan their work, take notes about what they plan to write, or store details they discover. The note and notebook functions are very useful for students to keep track of information or comment on peers' work. The information offered for each area and the writing prompts would interest students of primary and elementary ages, although many prompts might be a bit challenging for primary students to read independently. *UWCC* scores adequately on *Interface Design*, as primary students would be comfortable using this interface.

KWD also scored highest of the storybook programs on *Content*. The stated purpose of the storybook *Desktop Publishing Software* programs is to improve student writing by promoting the writing cycle and offering a fun atmosphere in which to write individually or collaboratively. These software programs facilitate the drafting and publishing processes for students, rather than directly instructing students in how to write effective stories. Although *EBD* and *SBW* perform the proofreading process for students through checking spelling, *KWD* does not have a spell-checking feature. Prewriting is possible in each program if students use pages to record ideas but the programs do not accommodate prewriting easily. Specifically, no space is provided for webbing, organizing ideas, or creating outlines. Collaborative writing is possible but not directly accommodated. Students could write a story together, but this is not a specific function built into the programs.

Unlike the storybook programs, *UWCC* is specifically designed to assist students with each aspect of the writing process through explaining the writing process and offering ideas for how to improve writing. The program offers ideas for students who need prompts for writing fiction and non-fiction, facilitates the organization of information, and encourages students to collaborate and assist their peers. These functions contribute to the higher *Content* score for *UWCC* than for the storybook programs which do not explain either the writing process or the purpose of the writing process.

None of the reviewed *Desktop Publishing Software* programs track student progress over time or allow for directed feedback; a major factor in how Bishop and Santoro (2006) separate educational software from non-educational software, so these programs are considered non-instructional. Unlike *TELE-Web*, which Englert, Manalo and Zhao (2004) reviewed, these programs do not provide assistance to students for formatting their work. In fact, the storybook programs provide neither directed feedback nor help with the writing process, so each of these programs scores below standard on *Instructional Design. UWCC* met expectation on *Instructional Design* because it more closely approximated how students might organize their work for Language Arts and Science in a primary classroom, and although it did not offer help formatting writing, it did offer direct assistance and strategies to help students understand and use the writing process.

Question 3: Are reading and writing prerequisite skills for using the software programs? "Writing is a recursive process. By following the writing process of prewriting, drafting, proofreading, and publishing, students learn to think and act as writers" (Creative Media Applications, 1997, p. 1), and the writing process, primarily publishing, is the primary concern of each of the software programs evaluated for this software category. Students use *EBD, KWD*, and *SWD* to create printable storybooks incorporating both text and graphics, and *UWCC* offers several types of pages such as a column layout for newsletters. Similar to most desktop publishers, each of these programs provide space for inputting text and enable modification of font type, size, colour, and alignment to varying degrees in order to suit personal preferences. Students must write in order to use the programs or they will not have a storybook or other written product to share with their audiences.

Unlike programs such as *K2*, where students organize ideas and then see these ideas transferred into an outline format to assist them in planning their writing, none of the storybook *Desktop Publishing Software* programs reviewed in this section offer a similar function to help students generate and organize ideas prior to writing. Students begin with a blank page at the drafting stage of the writing process and no space is provided for prewriting activities such as concept mapping. No ideas for how to write stories are offered and the programs do not either check or prompt students to check to ensure that story elements such as setting or conflict are present, that words form complete sentences, or that sentences form paragraphs, for example.

UWCC offers the notebook function and idea prompts to assist in idea generation. Drafting occurs on the page, and sticky notes allow peers to place notes on others' work when helping to revise. Students use tools such as the spellchecker to edit their work and the theatre to publish and share their work with others. At each stage of the writing process, the program explains what students should be doing and offers suitable ideas for how to improve student writing, whether individually or collaboratively. For example, brainstorming is suggested as a strategy for pre-writing, and the program explains brainstorming and offers suggestions how to do so effectively such as writing down all ideas, carrying out research, or talking to peers about the topic.

Reading is not a required skill for using the *Desktop Publishing Software* programs reviewed in this section, however the writing prompts offered by some of the programs, most notably *UWCC*, require students to read the prompt and write the story or non-fiction piece. *KWD* mentions the language-experience approach in its educational claims and students could read their own or their peers' writing, although reading is secondary to the writing function of the programs. In addition, each program saves stories and reads them through text-to-speech functions, although the pronunciation of words is often choppy and very fast. *KWD* permits students to input phonetic pronunciations to correct how the text-to-speech function pronounces words. The text-to-speech function could be used to improve students' reading fluency through reading with the program or recording student voices while reading the stories, but this reading activity is not directly addressed as a primary function of the software.

Question 4: Are these software programs appropriate and useful to supplement classroom reading and writing instruction? Each of the programs evaluated for this software category have a non-instructional focus and can be said to facilitate, rather than directly teach, writing skills development, as the responsibility is placed on teachers or other individuals to assist students to improve writing skills. *UWCC* offers the greatest amount of support for the writing process out of the four programs reviewed for *Desktop Publishing Software*.

EBD documentation (Creative Media Applications, 1997) states that the program is designed for students in third through eighth grades, *SBW* is listed as suitable for kindergarten through fifth grade (Riverdeep Interactive Learning, 2007), and *KWD* is suitable for pre-kindergarten through fourth grade (Carnow & Ellman, 2001). These software programs are appropriate for students in the primary grades, however older students may not find the programs appealing because the interface, topics, and graphics are more suited to primary students. *UWCC* is listed for students from first to fourth grade. Although they contain interesting facts about each "land" that would interest students of all ages, students in the first grade might find that the level of language used for the writing prompts is difficult to read.

A major emphasis of *Desktop Publishing Software* programs that is not apparent in *Word Processing Software* programs is the use of graphics to enhance presentation. With the exception of *KWD*, the addition of graphics into students' written work is simply for cosmetic purposes to enhance the overall presentation and appeal of the product for the publishing stage of the writing process, but graphics are not used in a way that would teach reading and writing. The *EBD* and *SWD* incorporate graphics into stories through the use of pre-generated backgrounds and "stamps" to place objects in the foreground. The stamped objects may be moved or slightly modified for size, colour, and placement on the page. *KWD* uses "stickers" in a similar manner, although the stickers may also be placed directly into stories to replace their label words. For example, students may replace *bird* with the stamp showing the picture of "bird", hear the computer-generated voice read *bird* for either the label or picture, and use the text label and picture interchangeably in the story. Replacing difficult words with pictures may help younger students to read more difficult stories with confidence as they develop their reading skills. The initial library of stickers is limited, but students may create their own stickers and link them to text labels. *UWCC* also provides a library of images and animations for use in stories, but does not provide a stamp or sticker function.

All of the *Desktop Publishing Software* programs reviewed are ten or more years old, and although they have been updated since their original publishing dates, their use of media is not as sophisticated as modern programs. Buttons are provided for common functions and are often accompanied by either a printed description such as *SWD* uses, or an auditory description such as *KWD* uses, so primary students would feel comfortable using the programs, but modern programs such as *K2* more closely integrate multimedia functions into program functions. *EBD*, particularly, has very pixelated graphics, which causes large squares rather than fine lines to appear on the screen, and would be less appealing to students who are more accustomed to high-resolution graphics.

Discussion of Software Usefulness for Primary Grades Instruction

Several issues appear to impact the usefulness of the software reviewed in this chapter for the purposes of reading and writing instruction in the primary grades. Many of the programs listed as authorized resources for reading and writing are not directly instructional or designed for teachers to evaluate student progress in the targeted skills, and may not be considered to help teach reading or writing. In addition, often programs simply replace the same functions performed by pencil and paper, or do not take advantage of the capabilities of modern computers. The findings for each research question discussed in previous sections raise issues about how the software programs are selected. These issues are discussed in the final section of this chapter which follows next.

Question 5: What issues or trends influence software selection and use? The first issue affecting the usefulness of the reviewed software is the characteristics of the programs themselves. Bishop and Santoro (2006) argue that for software to be considered educational it must save student work, offer specific feedback to allow students to improve their skills, track progress in a meaningful way so that teachers can make informed educational decisions to address areas of need, and use progress indicators to adapt program functioning to meet student needs. For example, if a program began by addressing short vowel consonant-vowel-consonant (cvc) words, and students were having difficulty, a good educational program would offer feedback and extra practice before allowing students to progress to the next skill. If students had mastered short-vowel cvc words, the program would automatically progress to the next skill in sequence. Aside from the *Drill and Practice Games*, none of the reviewed software offers automatic feedback for reading and writing skill development. In many cases, reading and writing skills are indirectly addressed, for instance, several word processing and desktop publishing programs claim to promote reading fluency because students are able to record their own voices reading their text or read along with the computer-generated voice using the text-to-speech functions. These functions are optional and are not part of the primary functioning of the program. No evaluative function exists for the reviewed programs, with the exception of *DBS* and *WOS*. It was expected that the *Drill and Practice Games* would contain some sort of evaluative function to track student achievement as they progressed through each activity, but the programs did not have this feature either. Perhaps, as these programs were not designed exclusively for school use, the evaluative function was excluded. In light of Bishop and Santoro's (2006) contention that educational software should track student progress and provide directed feedback, the majority of the reviewed authorized programs would not be considered good educational software.

Few of the reviewed writing programs, with the exception of *UWCC*, actually address writing skills or the writing process. For the most part, these programs are tools students use to assist them as they write, but unlike Englert, Manalo, and Zhao's statement, "I can do it better on the computer" (2004, p. 5), the writing programs typically replace the same types of activities that were traditionally done with pencil and paper. Concept maps, like other graphical organizers, may be drawn by hand and, in some cases, drawing by hand is faster and more efficient. Students incorporate pictures, sounds, animations, and links to other documents or the internet into the concept maps created on the computer, and although this might help them to organize ideas more effectively, the main difference between the software and the traditional pencil and paper method is simply the visual appeal of the final product. One function performed by the software that traditional methods cannot perform is converting concept maps into linear outline format so that students may use the outlines as tools for drafting their work. *Word Processing Software* and *Desktop Publishing Software* do not offer the kinds of support for writing structure that Englert, Manalo, and Zhao (2004) and Wade-Stein and Kintsch (2004) found in programs designed to directly teach narrative and expository paragraph structures because the programs reviewed for my thesis were found to be not specifically designed to teach writing skills. Even the included tools, such as spell checkers or dictionaries, simply replace their non-digital counterparts and do not teach spelling, for example.

Several obstacles hinder technology integration in the classroom when using the software reviewed for this thesis. Most of the writing software programs are non-instructional tools which fits Page's view of technology integration (2002), where technology is used as simply tools for learning. However, the increased time to create templates for students and secure adequate computer access for all students supports Bauer and Kenton's (2005) concern that many teachers perceive that it requires more planning and implementation time to integrate technology into instruction than to use traditional instructional methods. In many situations, for example, students draft their work with pencil and paper and then use the computer to type a final draft, rather than drafting and revising their work with software. This practice is neither consistent with the vision of integration that Page presents, nor with provincial and territorial integration requirements.

The final barrier to meaningful technology integration using the listed software is the actual list itself. The western provinces (including Manitoba, Saskatchewan, Alberta, and British Columbia) and the northern territories (including the Yukon Territory, the Northwest Territories, and Nunavut) collaboratively chose software that fit a more constructivist viewpoint of learning, rather than Drill and Practice Games (see, for example, Western Canadian Protocol for Collaboration in Education, 2002). With the exception of the Yukon Territory and British Columbia, these software lists have not been updated and contain some titles no longer available from their publishers, such as *Has Anybody Seen My Umbrella?* The software on these lists may not accurately reflect the software currently in use in the western provinces and northern territories yet no current options are offered for purchase by provincial and territorial Ministries of Education to guide school and district software purchases. Newer programs available in the eastern provinces (including Nova Scotia, New Brunswick, Newfoundland and Labrador, and Prince Edward Island), such as *Draft: Builder* which is listed for Nova Scotia, are not listed as available for purchase from the Learning Resources Distribution Centre in Alberta, for example (Alberta Education, 2008b). Most of the eastern provinces have similar lists, however, Newfoundland and Labrador and New Brunswick do not have central lists (see Appendix A).

The listed software for Nova Scotia and Prince Edward Island, such as *Kidspiration 2*, *Appleworks*, and *Co: Writer* are mostly tools for writing, and are non-instructional in focus. Schools and districts in Eastern Canada looking for guidance on educational games to help teach reading or writing skills, for example, will not find this guidance on their central lists. Ontario has the most extensive list of software titles but the province licenses a mixture of new and old programs, rather than listing any of the software programs as authorized.

Based on my systematic analysis of the authorized software for use in primary grades, two problems are clear. First, the software listed for the provinces and territories are not helpful tools for the integration of computers and computer-assisted instruction. Second, the programs lack educational support through tracking student progress and providing feedback, do not teach skills in context, are assistive rather than instructional, and are often outdated. Thus, for integration to be accomplished, a more thoughtful consideration of available programs, their benefits to students, and how they align with the Programs of Study is needed.

CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS

The software review presented in this thesis reflects an appraisal of the authorized software for teaching reading and writing across Canada. Although not all of the programs on the lists for all provinces and territories were reviewed, those software programs reviewed represent the types of programs typically available.

Concluding Remarks Concerning the Research Questions

When considering what types of authorized software are available in Canada the following was determined:

- The authorized software could be organized into four main groups: *Reading Programs*, including Drill and Practice Games and Talking Books, *Writing Programs*, including Concept Mapping Software, Word Processing Software, and Desktop Publishing Software, *Assistive Programs for Students with Special Needs*, and *Other Programs* linked to paper resources.
- Overlap exists between some provincial and territorial software lists, especially for the western provinces and northern territories, but no consistency exists across all provincial and territorial jurisdictions.
- Some provinces, such as New Brunswick and Newfoundland and Labrador do not have central lists of authorized software.
- Many listed programs are outdated and unavailable for purchase.

Software manufacturers make varied claims regarding the educational benefits of using their products. Often:

- The claims are often broad and no evidence is provided for how they are met, such as reporting educational trials.

- Claims include educational vocabulary or refer to educational approaches, such as the language-experience approach for reading fluency and comprehension development.
- Programs adequately support their main educational claims through activities but the non-specific nature of the claims is problematic and secondary claims are often indirectly supported.

After determining whether the sample of authorized software programs actually require students to read or write in order to use the program, it was determined that:

- Software designed for children in the primary grades often includes text-to-speech functions and graphical interfaces to assist students with reading.
- Drill and Practice Games designed for the primary grades may not require students to read in order to play the game because gradual elimination is used in some programs.
- Reading programs focus on reading skills and often do not support writing skills such as drafting responses.
- Writing programs require writing to be used in most cases and indirectly support reading because students read directions or their own work.

When determining whether the authorized programs are appropriate or useful tools for supplementing classroom instruction, it was concluded that:

- Most programs listed as authorized software to teach reading and writing in the primary grades are suitable for use by primary students.
- Most listed programs are not directly instructional and may be considered tools.

- Drill and Practice Games permit students to practice reading skills but do not permit teachers to easily monitor student progress.
- Writing programs typically replace functions traditionally performed manually by students and do not teach the writing process.

Many issues and trends influence what software is authorized for use by provincial and territorial Ministries of Education. The major issues affecting the use of authorized software in classrooms were:

- Software selected as authorized resources should fit the educational goals set forth the in the Programs of Study and reflect current teaching practices, but this is not the case in many instances.
- Lists containing outdated software may not reflect current programs that are being used in Canadian schools.

Implications for Policy Decisions

To effect the curricular expectation for meaningful integration of computer technology into other curricular areas, the first and most important step, whether at the provincial and territorial level or at the national level, would be to examine the Language Arts and Technology Programs of Study and thoroughly and thoughtfully consider which current resources are available that would both act as tools for students and as instructional aids to help teach reading and writing in ways that are consistent with current educational practices and curricula. Such a list would be extensive, and would give educators, schools, and school districts guidance on what types of criteria were used in the selection process so that they can choose resources from this list, or from other sources that would align with the aims of the Programs of Study. This list would require updating as new software was designed that addressed content in different ways. In addition to the list, these resources would need to be available for purchase from the provincial or territorial resource distribution centres or licensed in a similar manner to Ontario's Educational Software Service so that they are available to all schools.

Recommendations For Further Research

The software reviewed in this thesis represents a limited selection of the programs listed as authorized resources for teaching reading and writing in the Canadian provinces and territories. A more extensive review of the software with each provincial or territorial Language Arts and Technology Program of Studies would be required to determine exact curricular fit or usefulness of each program listed as an authorized resource for teaching reading or writing. An in-depth study of the authorized programs to determine whether they actually require reading and writing skills to be used is also recommended. In addition, a survey of what types of programs are actually in use in Canadian schools, and a needs assessment to determine what types of software teachers would consider useful to fit with curricular expectations and their own teaching would be a beneficial starting point.

Limitations of My Study

Two limitations affecting this study are:

- Only 13 programs were available for review due to limited availability of the software and amount of outdated titles.
- Many schools and districts have adopted site-based management and decision-making, so it is conceivable that other programs beyond the programs listed as

authorized programs are being used in Canadian schools, however determining which programs are being used in all Canadian schools is beyond the scope of my thesis.

References

Alberta Education (2008a). *Alberta Resources Catalogue*. Edmonton, AB: Author. http://www.lrc.education.gov.ab.ca/pro/cat/cat_alberta.htm.

Alberta Education (2008b). *Authorized Resources Database*. Edmonton, AB: Author. http://education.alberta.ca/apps/lrdb/

Alberta Learning (2000). *Information and Communication Technology.* Edmonton, AB: Author. http://education.alberta.ca/media/453069/pofs.pdf

Bauer, J., & Kenton, J. (2005). Toward technology integration in the schools: Why it isn't happening. *Journal of Technology and Teacher Education, 13*(4), 519-546.

Becker, H. J. (1998). Running to catch a moving train: Schools and information technologies. *Theory into Practice, 37*(1), 20-30.

Bishop, M. J., & Santoro, L. E. (2005). *Early Reading Software Evaluation Form*. PA: Authors. http://www.lehigh.edu/~mjba/TABR/pdf/ReadingSWInstrument _090205.pdf.

Bishop, M. J., & Santoro, L. E. (2006). Evaluating beginning reading software for at-risk learners. *Psychology in the Schools, 43*(1), 57-70.

British Columbia Ministry of Education. (2007). *Curriculum*. Victoria, BC: Author. http://www.bced.gov.bc.ca/irp/

British Columbia Ministry of Education. (2008). *British Columbia Learning Resources Catalogue 2008*. Victoria, BC: Author. http://www.lrc.education.gov.ab.ca/rcbc/ BC_1_Recommended/English/Elementary.pdf.

Butzin, S. (2001). Using instructional technology in transformed learning environments: An evaluation of project child. *Journal of Research on Computing in Education, 33*(4), 367-373.

Carnow, G., & Ellman, B. (2001). *Kid Works Deluxe: Software, Instructions, and Standards-Based Lessons*. Westminster, CA: Teacher Created Materials, Inc.

Connell, J. E., & Witt, J. C. (2004). Applications of computer-based instruction: Using specialized software to aid letter-name and letter-sound recognition. *Journal of Applied Behaviour Analysis, 37*(1), 67-71.

Creative Media Applications (1995). *A to Zap!: Teacher's Guide.* Elgin, IL: Sunburst Technology.

Creative Media Applications (1997). *Easy Book Deluxe: Teacher's Guide.* Elgin, IL: Sunburst Technology.

Dwyer, J. (2007). Computer-based learning in a primary school: Differences between the early and later years of primary schooling. *Asia-Pacific Journal of Teacher Education, 35*(1), 89-103.

Englert, C. S., Manalo, M., & Zhao, Y. (2004). I can do it better on the computer: The effects of technology-enabled scaffolding on young writers' composition. *Journal of Special Education Technology, 19*(1), 5-22.

Fasting, R. B., & Lyster, S. H. (2005). The effects of computer technology in assisting the development of literacy in young struggling readers and spellers. *European Journal of Special Needs Education, 20*(1), 21-40.

First Peoples' Cultural Foundation. (2007). *First Voices.* Brentwood Bay, BC: Author. http://www.firstvoices.ca/

Franklin, C. (2007). Factors that influence elementary teachers use of computers. *Journal of Technology and Teacher Education, 15*(2), 267-293).

Gill, S. R. (2007). Learning about word parts with kidspiration. *Reading Teacher, 61*(1), 79-84.

Judge, S., Puckett, K., & Bell, S. M. (2006). Closing the digital divide: Update from the early childhood longitudinal study. *Journal of Educational Research, 100*(1), 52-60.

Lefever-Davis, S., & Pearman, C. (2005). Early readers and electronic texts: Cd-rom storybook factors that influence reading behaviours. *The Reading Teacher, 58*(5), 446-454.

Littleton, K., Wood, C., & Chera, P. (2006). Interactions with talking books: Phonological awareness affects boys' use of talking books. *Journal of Computer-Assisted Learning, 22*(5), 382-390.

Manitoba Textbook Bureau. (n.d.). *MTBB Home*. Souris, MB: Author. http://www.mtbb.mb.ca/catalogue/en/.

McVee, M. B., & Dickson, B. A., (2002). Creating a rubric to examine literacy software in the primary grades. *The Reading Teacher, 55*(7), 635-639.

Mostow, J., Aist, G., Burkhead, P., Corbett, A., Cuneo, A., Eitelman, S., et al. (2003). Evaluation of an automated reading tutor that listens: Comparison to human tutoring and classroom instruction. *Journal of Educational Computing Research, 29*(1), 61-117.

Norris, C., Sullivan, T., Poirot, J., & Soloway, E. (2003). No access, no use, no impact: Snapshot surveys of educational technology in k-12. *Journal of Research on Technology in Education, 36*(1), 15-27.

Nova Scotia School Book Bureau. (n.d.). *Nova Scotia School Book Bureau On-Line System*. Halifax, NS: Government of Nova Scotia. https://w3apps.ednet.ns.ca/nssbb/

Ontario Educational Software Service. (2008). *OESS Data Base*. Toronto, On: Ontario Software Acquisition Program Advisory Committee. http://www.osapac.org/dbOESS/dbOESSe1.asp

Page, M. S. (2002). Technology-enriched classrooms: Effects on students of low socio-economic status. *Journal of Research on Technology in Education, 34*(4), 389-409.

Prince Edward Island Department of Education. (2004). *Journey On: Continuing to Integrate Information Technology in PEI Schools*. Charlottetown, PE: Author. http://www.edu.pe.ca/journeyon/index.htm

Prince Edward Island Department of Education (2006). *Elementary Program of Studies and Authorized Materials 2006-2007*. PEI: Author. http://www.gov.pe.ca/photos/original/ed_eps_0708.pdf.

Riverdeep Interactive Learning (2007). Storybook weaver deluxe 2004. *Riverdeep, Destination Success, Reading Software, Math Software*. San Francisco, CA: Author.http://web.riverdeep.net/portal/page?_pageid=818,1384409,818_1384439&_dad=portal&_schema=PORTAL

Saskatchewan Education. (2007). *Learning Resources.* Regina, SK: Author. http://www.learning.gov.sk.ca/Default.aspx?DN=e8f55828-b8e2-4c79-b1b1-8f911043ffe0

Sim, G., MacFarlane, S., & Read, J. (2006). All work and no play: Measuring fun, usability and learning in software for children. *Computers & Education, 46*, 235-248.

Sorrell, C. A., Bell, S. M., McCallum, R. S. (2007). Reading rate as a function of computerized versus traditional presentation mode: A preliminary study. *Journal of Special Education Technology, 22*(1), 1-12.

Statistics Canada. (2006a). *Characteristics of Individuals Using the Internet.* Ottawa, ON: Author.http://www40.statcan.ca/l01/cst01/comm14a.htm?sdi=school.

Statistics Canada. (2006b). *Internet Use by Individuals, By Location of Access, By Province*. Ottawa, ON: Author. http://www40.statcan.ca/l01/cst01/comm15.htm?sdi=children.

Wade-Stein, D., & Kintsch, E. (2004). Summary street: Interactive computer support for writing. *Cognition and Instruction, 22*(3), 333-362.

Wozney, L., Venkatesh, V., & Abrami, P. C. (2006). Implementing computer technologies: Teachers' perceptions and practices. *Journal of Technology and Teacher Education, 14*(1), 173-207.

Western and Northern Canadian Protocol. (2008). *Western Canadian Protocol Site.* Edmonton, Alberta: Author. http://www.wncp.ca/.

Western Canadian Protocol for Collaboration in Education. (2002). *Kindergarten to Grade 10 English Language Arts Resources: Compilation of Annotated*

Bibliographies from 1997 to 2001. The Governments of Manitoba, Saskatchewan, Alberta, British Columbia, Yukon Territory, and the Northwest Territories. http://www.edu.gov.mb.ca/k12/learnres/wncp/wcpelak-10com_suppl.pdf.

Yukon Department of Education. (2007). *Yukon K-7 ICT Curriculum: Information and Communication Technology*. Whitehorse, YT: Author. http://dl1.yukoncollege.yk.ca/ K7ICT/.

Yukon Education Student Network. (2008). *IT Resources Software List*. Whitehorse, YT: Author. http://www.yesnet.yk.ca/tal/it_softwareres.html

Appendices

Appendix A: Software Titles by Province

Province	Title	Availability and Notes
Alberta	• *Chicka Chicka Boom Boom (1996)* • *Easy Book Deluxe (1998)* • *Kid Works Deluxe (1995)*	• Resource list available at http://www.lrc.education.gov.ab.ca/pro/cat/cat_alberta.htm • Or on the searchable site: http://education.alberta.ca/apps/lrdb/
British Columbia	• *Collections 2 and 3 –audio pack (1999)* • *Easy Book Deluxe (1998)* • *Kid Works Deluxe (1995)* • *Literacy Place for the Early Years – books and CD-ROM* • *Ultimate Writing and Creativity Centre (1997)*	• The curriculum documents with resource list available at http://www.bced.gov.bc.ca/irp/ • Another list of recommended resources may be found at http://www.lrc.education.gov.ab.ca/rcbc/BC_1_Recommended/English/Elementary.pdf
Manitoba	• *Chicka Chicka Boom Boom (1996)* • *Easy Book Deluxe (1998)* • *Has Anybody Seen My Umbrella? (1998)* • *Kids Works Deluxe CD-ROM (1995)* • *Nelson Spelling 3 CD-ROM (1997)* • *Ultimate Writing and Creativity Centre (1997)*	• Western Canadian Protocol resources available at http://www.edu.gov.mb.ca/k12/learnres/wncp/wcpelak-10com_suppl.pdf • With other resources being able to be ordered from the Textbook Bureau: http://www.mtbb.mb.ca/catalogue/en/

New Brunswick		• No master list available for the Province of Newfoundland and Labrador. Schools and districts choose their own programs.
Newfoundland and Labrador		• No master list available for the Province of Newfoundland and Labrador. Schools and districts choose their own programs.
Nova Scotia	• *Amazing Writing Machine (1997)* • *Community Construction Kit* • *Co:Writer 4000 (2004)* • *Draft: Builder (2004)* • *Inspiration (2006)* • *KidSpiration (2005)* • *Storybook Weaver Deluxe (1996)*	• Able to be ordered from: https://w3apps.ednet.ns.ca/nssbb/
Nunavut	• *Two Titles in Inuktitut*	• Language instruction from Kindergarten to grade Three occurs in Inuktitut in most schools in Nunavut.
Northwest Territories	• *Chicka Chicka Boom Boom (1996)* • *Easy Book Deluxe (1998)* • *Has Anybody Seen My Umbrella? (1998)* • *Kid Works Deluxe (1995)*	• No titles listed. Teachers are recommended to use the Western Canadian Protocol resources available at http://www.wncp.ca/
Ontario	• *A to Zap! (1998)* • *ABCircus (2000)*	• No resources listed as Approved or Recommended, however they license these titles for use in all

	• *Clicker 4 (2004)* • *Clicker 5 (2006)* • *Co:Writer 4000 (2004)* • *Co:Writer 4000 Solo (2005)* • *EasyBook Deluxe (1998)* • *Easybook Deluxe (2006)* • *House Series (Bailey's) (2001)* • *Kid Pix Deluxe 3 (2002)* • *Kid Pix® Deluxe 4 (2005)* • *Microsoft Publisher (2000) (2002)* • *Reader Rabbit 1 and Deluxe (1997)* • *Reader Rabbit 2 and Deluxe (1997)* • *Reader Rabbit 3 and Deluxe (1997)* • *Sitting on the Farm (1996)* • *SMART Ideas ® Concept Mapping Software (2005)* • *SMART Ideas® software 5.0 (2005)* • *Star Office 7 Office Suite (2004)* • *Storybook Weaver Deluxe (1996)*	schools. Resources are listed here: http://www.osapac.org/dbOESS/dbOESSe1.asp

	• *The Cat Came Back (1996)* • *The Print Shop® for Mac (7 & 8) (2005)* • *WORDville Word Way (2001)* • *WORDville Write Way (2001)* • *Write:Outloud (2004)* • *Write:Outloud Solo (2005)*	
Prince Edward Island	• *Appleworks* • *Dragon Naturally Speaking* * • *Inspiration 7.5* • *Kurzweil* * • *Rosetta Stone* * • *Ultimate Writing and Creativity Centre (1997)*	• Resources are listed here: http://www.edu.pe.ca/journeyon/index.htm
Québec		• Language instruction occurs in French in most schools in Québec.
Saskatchewan	• *Changes: The Turtle's Teachings (2001)* • *Chicka Chicka Boom Boom (1996)* • *Has Anybody Seen My Umbrella? (1998)*	• Resources are listed here: http://www.learning.gov.sk.ca/Default.aspx?DN=e8f55828-b8e2-4c79-b1b1-8f911043ffe0.
Yukon Territory	• *Appleworks* • *Comic Life (2004)*	• Resources are listed here: http://www.yesnet.yk.ca/tal/it_softwareres.html

- *First Voices*
- *Inspiration 8 (2006)*
- *Kid Pix Studio Deluxe 4 (2005)*
- *Kidspiration 2 (2005)*
- *ThinkFree Office*

* Designated for special needs students

Appendix B: Alphabetical List of Titles with Publishers

Title	Publisher	Type	Province
A to Zap! (1998)	PCI Education	Drill and Practice Game – Letters	9
ABCircus (2000)	CatEdu Math & Science Software	Drill and Practice Game – Letters	9
Amazing Writing Machine (1997)	Riverdeep Interactive Learning (Formerly: The Learning Company)	Word Processor	6
Appleworks	Apple	Word Processor	10, 13
Changes: The Turtle's Teachings (2001)	Gabriel Dumont Institute of Native Studies	Talking Books – Audio CD accompanying a book	12
Chicka Chicka Boom Boom (1996)	Davidson and Associates Inc.	Interactive Talking Book, Drill and Practice Game - Letters	1, 3, 8, 12
Clicker 4 (2004)	Crick Software	Word Processor – Voice Recognition Software	9
Clicker 5 (2006)	Crick Software	Word Processor – Voice Recognition Software	9
Co:Writer 4000 (2004)	Don Johnson Incorporated	Word Prediction Software	6, 9
Co:Writer 4000 Solo (2005)	Don Johnson Incorporated	Word Prediction Software	9
Collections 2 and 3 – Audio Pack (1999)	Pearson Education Canada	Talking Book, Program Support to Printed Materials	2

Comic Life (2005)	Plasq	Desktop Publishing	13
Community Construction Kit	Tom Snyder Productions	Creative Writing	6
Draft: Builder	Don Johnson Incorporated	Writing Process Assistance	6
Dragon Naturally Speaking	Nuance	Voice Recognition Software	10
Easy Book Deluxe (1998) (2006)	Sunburst Technology	Word processor, Desktop Publishing	1, 2, 3, 8, 9
First Voices	Web-based: http://www.firstvoices.ca/	First Nations Language and Culture Website	13
Has Anybody Seen My Umbrella? (1998)	National Film Board of Canada	Interactive Talking Book, Drill and Practice Game	3, 8, 12
House Series (Bailey's) (1995)	Riverdeep Interactive Learning (Formerly: Edmark Corporation)	Drill and Practice Game, Creative Writing	9
Inspiration 7.5	Inspiration Software Co.	Concept Mapping	10
Inspiration 8 (2006)	Inspiration Software Co.	Concept Mapping	6, 13
Kid Pix Deluxe 3 (2002)	Riverdeep Interactive Learning (Formerly: Broderbund)	Desktop Publishing	9
Kid Pix Studio Deluxe 4 (2005)	Riverdeep Interactive Learning (Formerly: Broderbund)	Desktop Publishing	9, 13
Kid Works Deluxe (1995)	Davidson and Associates Inc.	Word processor, Desktop Publishing	1, 2, 3, 8
Kidspiration	Inspiration Software Co.	Concept Mapping	6
Kidspiration 2 (2005)	Inspiration Software Co.	Concept Mapping	13

Kurzweil	Kurzweil Educational Systems	Text-to-Speech, Voice Recognition Software	10
Literacy Place for the Early Years – books and CD-ROM	Scholastic	Program Support to Printed Materials	2
Microsoft Publisher (2000) (2002)	Microsoft	Desktop Publishing	9
Nelson Spelling 3 CD-ROM (1997)	Nelson Education	Program Support to Printed Materials	3
Reader Rabbit 1 and Deluxe	Riverdeep Interactive Learning (Formerly: The Learning Company)	Drill and Practice Game	9
Reader Rabbit 2 and Deluxe	The Riverdeep Interactive Learning (Formerly: The Learning Company)	Drill and Practice Game	9
Reader Rabbit 3 and Deluxe	Riverdeep Interactive Learning (Formerly: The Learning Company)	Drill and Practice Game	9
Rosetta Stone	Rosetta Stone Inc.	Voice Recognition Software	10
Sitting on the Farm (1996)	Focus Multimedia Softare	Interactive Talking Book, Creative Writing	9
SMART Ideas ® Concept Mapping Software (2005)	SMART Technologies	Concept Mapping	9
SMART Ideas® Software 5.0 (2005)	SMART Technologies	Concept Mapping	9
Star Office 7 Office Suite (2004)	Sun Microsystems	Word Processor	9

Storybook Weaver Deluxe (1996)	Riverdeep Interactive Learning (Formerly: The Learning Company)	Desktop Publishing	6, 9
The Cat Came Back (1996)	Focus Multimedia Software	Interactive Talking Book, Creative Writing	9
The Print Shop® for Mac (7 & 8) (2005)	Riverdeep Interactive Learning (Formerly: Broderbund)	Desktop Publishing	9
ThinkFree Office (2007?)	Haansoft ThinkFree Co. Ltd	Word Processor	13
Ultimate Writing and Creativity Centre (1997)	Riverdeep Interactive Learning (Formerly: The Learning Company)	Word Processor	2, 3, 10
WORDville Word Way (2001)	Courseware Solutions Inc.	Drill and Practice Games – General Language Skills	9
WORDville Write Way (2001)	Courseware Solutions Inc.	Drill and Practice Game – General Language Skills and Writing skills	9
Write:Outloud (2004)	Don Johnson Incorporated	Word Processor	9
Write:Outloud Solo (2005)	Don Johnson Incorporated	Word Processor	9

Province Codes

Alberta – 1
British Columbia – 2
Manitoba – 3
New Brunswick – 4
Newfoundland and Labrador – 5
Nova Scotia – 6
Nunavut – 7
Northwest Territories – 8
Ontario – 9
Prince Edward Island – 10
Québec – 11
Saskatchewan – 12
Yukon Territory – 13

Appendix C: Procedures for Using the Evaluation Instrument

(summarized and adapted from Bishop and Santoro (2006))

Interface Design:

1. Consider the program without the instruction manual.
2. Launch the title sequence, allowing it to play through uninterrupted.
    a. How long is it? Can it be bypassed?
3. Quit the program and relaunch it bypassing the title screen.
4. Log in or create a log in account if prompted.
    a. If you are not asked to log in, the program does not track student performance and growth over time.
    b. If you are asked to log in, can an administrator make accounts before use instead of requiring students to make accounts individually?
5. Examine the program interface.
    a. What are the quality of the fonts, graphics and music?
    b. Are the screens well-organized and of consistent design throughout?
    c. Can the screens be rearranged to fit user preferences or needs?
6. Check to see whether instructions are included both in writing and orally.
    a. Are instructions repeatable?
7. Deliberately make a mistake in navigating through the program.
    a. Does the program simply make a beep or sound to indicate errors, or is help context sensitive?
    b. Repeat the error several times. Does the program eventually show you what to do, rather than explain the correct procedure?

8. Examine the buttons and screen lay out. Is the interface user friendly?
9. Determine the level of interactivity.
    a. Is more time spent on watching the screen or actively learning?
    b. Do interactions between the user and the program alter the way the program functions?

Content Delivery and Instructional Design:

1. Examine the teacher's material.
    a. What does the designer say that the program should do?
    b. What skills are supposedly developed?
2. Run the program and try each part.
    a. Are the skills the publisher maintains are being taught actually practiced?
3. Decide how systematically the program handles content.
    a. Are skills presented in sequence from easy to more difficult?
    b. Does the program model skills before having students practice material?
    c. What supports and scaffolds are in place to assist learners? Is feedback specific and targeted to the response? Is feedback flexible or prescriptive?
4. Attempt to leave the program in the middle of a learning task.
    a. Does the program restart, or continue from where it was?
5. Determine what types of feedback are available.
    a. Are students able to see the evaluation of their progress?
    b. Is the feedback available and useful for teachers?
    c. Does the program use feedback on student progress to modify program delivery to meet student needs?

Appendix D: Evaluation Instrument

(adapted from Bishop and Santoro (2005), found on-line at: http://www.lehigh.edu/~mjba/TABR/pdf/ReadingSWInstrument_090205.pdf)

Background Information

Software name and version number:		
Publisher:	Year Published:	Cost: $
Operating System (Mac/Win/both) Evaluated under: Mac? Win?		
Type of Program:		Skills Developed: Reading/Writing/both
Package contents:		
Supplementary materials:		
Required materials:		
Stated goal and objectives:		Clearly defined? SD D N A SA
Stated target audience:		Clearly defined? SD D N A SA
Stated prerequisite skills:		Clearly defined? SD D N A SA

Interface Design: To what extent is the software...

Aesthetically Pleasing? Does the program's interface use media (text, graphics, animations, video, sound) in ways that enhance the experience?		
	The media used is high quality.	SD D N A SA
	Screens are laid out in well-organized ways (rather than haphazard placement of objects).	SD D N A SA
	Screens are neither overly stimulating nor boring.	SD D N A SA
	The "look and feel" of this program is likely to be pleasing to the learner.	SD D N A SA
	Media are used to create themes/metaphors that relate to the content and help create meaning.	SD D N A SA
	Learner is able to modify the interface according to individual preferences.	SD D N A SA
Supportive Operationally? Will the pre-reading learner be able to use the program with little help from adults?		
	Direct Support:	
	All operational instructions are supplied auditorially within the program.	SD D N A SA
	Operational instructions can be reviewed, as necessary.	SD D N A SA
	Instructions supplied within the program will be helpful to the intended audience	SD D N A SA
	The interface responds with prompt and informative invalid action messages when appropriate.	SD D N A SA
	After repeated invalid actions, the interface shows the learner how to correctly operate the function.	SD D N A SA
	Indirect Support:	
	The interface takes advantage of what learners already know.	SD D N A SA
	Learners don't have to search for commonly used functions.	SD D N A SA
	Program functions are placed in equivalent, if not identical, locations on screens.	SD D N A SA
	Things on the screen are what they appear to be and function as expected.	SD D N A SA
Interactive? Is the learner the primary driving force behind what happens in the program?		
	The learner rarely sits passively watching as the program does things.	SD D N A SA
	Interactions are frequent.	SD D N A SA
	The learner interacts directly with screen objects.	SD D N A SA
	Interactions with screen objects are as nearly like their real-world referents as possible.	SD D N A SA
	Learner interactions make a substantive difference in what the program is doing.	SD D N A SA

Content: To what extent does the software address the development of...

First major skill listed in the documentation:		
		SD D N A SA
		SD D N A SA
		SD D N A SA
		SD D N A SA
		SD D N A SA
Second major skill listed in the documentation:		
		SD D N A SA
		SD D N A SA
		SD D N A SA
		SD D N A SA
		SD D N A SA
Third major skill listed in the documentation:		
		SD D N A SA
		SD D N A SA
		SD D N A SA
		SD D N A SA
		SD D N A SA

Instructional Design: To what extent is the software...

Systematic? Is the instruction comprised of cycles that progress hierarchically through increasingly difficult blocks of content and skill sets?		
	The program gains learners' attention at the beginning of each instructional cycle.	SD D N A SA
	Learners are reminded of prerequisite knowledge at the beginning of each instructional cycle.	SD D N A SA
	The program informs learners of objectives at the beginning of each instructional cycle.	SD D N A SA
	The program offers multiple examples of a target skill (including use of pseudowords).	SD D N A SA
	The program supplies adequate opportunities for learners to practice newly learned skills.	SD D N A SA
	Learners must demonstrate mastery of previously introduced skills before moving on to new skills.	SD D N A SA
	The program supplies larger conceptual anchors for retention and retrieval (transfer of knowledge).	SD D N A SA
Instructionally Supportive? Does the program supply appropriate levels of content support to enhance learning?		
	The program makes content support available precisely when the learner needs it.	SD D N A SA
	The content support provided is helpful, but not so prescriptive that is short-circuits learning.	SD D N A SA
	The program uses informative, instantaneous feedback messages to support content learning.	SD D N A SA
	The program branches automatically to accommodate learner's remediation needs.	SD D N A SA
	The relevance of learning activities is made clear to the learner.	SD D N A SA
Assessing? Does the program evaluate learner progress and help direct learning goals?		
	The program saves learners' work	SD D N A SA
	The program supplies progress summaries.	SD D N A SA
	The program graphs or charts learner performance in an easily interpreted way.	SD D N A SA
	The program interprets learner performance and makes recommendations for how to proceed.	SD D N A SA
	The program includes an administrative function that tracks all learners working with it.	SD D N A SA

SUMMATIVE EVALUATION

Interface Design:	Score/ Percent
Content:	Score/ Percent
Instructional Design:	Score/ Percent
Stated goal and objectives were met:	SD D N A SA
Stated target audience was served:	SD D N A SA
Stated prerequisite skills were accurate:	SD D N A SA

COMMENTS:

Appendix E: List of Software Title Acronyms

Acronym	Full Software Title	Page Acronym is Introduced
AtZ	*A to Zap!*	*49*
BBH	*Bailey's Book House*	*49*
C5	*Clicker 5*	*68*
DBS	*Draft: Builder Solo*	*58*
EBD	*Easy Book Deluxe*	*74*
I8	*Inspiration 8*	*58*
K2	*Kidspiration 2*	*58*
KWD	*Kid Works Deluxe*	*74*
RR1	*Reader Rabbit 1*	*49*
SI5	*SMART Ideas 5.0*	*58*
SWD	*Storybook Weaver Deluxe*	*74*
UWCC	*The Ultimate Writing and Creativity Centre*	*74*
WOS	*Write: Outloud Solo*	*68*