JRTE, *42*(2), 197–216

Commercial Software Programs Approved for Teaching Reading and Writing in the Primary Grades: Another Sobering Reality

Meridith Lovell
Linda Phillips
University of Alberta

Abstract

This article reports the results of a systematic and comprehensive evaluation of the suitability of 13 commercially available, authorized software programs for teaching reading and writing in the primary grades. These programs were assessed on interface design, content, instructional design, whether manufacturers' educational claims were supported by the programs, and appropriateness to supplement reading and writing instruction. Regardless of date of publication, most software programs were judged to be noninstructional, in that they did not track student progress, provide feedback, or adapt to suit student needs. Many used decade-old interface design and program functions as well as content features, thereby limiting their usefulness as educational tools. (Keywords: computer-assisted instruction, computer integration, commercial software programs as teaching resources, reading instruction, writing instruction)

INTRODUCTION

The extent to which schools are keeping pace with the shift from conventional books to computer technology in the teaching of reading and writing is a critically important question. In their 2006 study, Judge, Puckett and Bell (2006) wrote, "… most educators agree that computer access and literacy have become vital and necessary for young learners in the 21st century" (p. 52). In fact, computer technology use is presumed to be commonplace in most western schools, and cited benefits include the option for teachers to individualize instruction (Englert, Manalo, & Zhao, 2004), accommodate students to work at their own pace (Littleton, Wood, & Chera, 2006; Sorrell, Bell, & McCallum, 2007), provide instantaneous feedback (Butzin, 2001; Fasting & Lyster, 2005), and manage recordkeeping functions (Bishop & Santoro, 2006).

Most ministries of education in Canada's provinces and territories have mandated technology integration across all subject areas. Accordingly, they provide lists of software programs designated as authorized resources. This paper reports on the results of a systematic and comprehensive evaluation of the suitability of commercially available, authorized software programs for teaching reading and writing in the primary grades. We provide and discuss the results of a detailed analysis of the congruence between the authorized technology and language arts curricula, the educational claims made by the software manufacturers and their match with the software, and the appropriateness of the software programs to supplement the teaching of reading and writing.

Background

Increasing emphasis on computer technology and literacy in classrooms reflects the accelerating importance of computer technology and literacy in society. Appropriate technology use and access to facilitate learning is an important aspect of contemporary education, especially as most children will frequently encounter and interact with technology outside the classroom. To prepare students to use technology as a tool for learning, problem-solving applications, and entry into the digital world, most technology programs of study require teachers to meaningfully integrate computer technology into the daily educational experiences of students (Alberta Learning, 2000; Prince Edward Island Department of Education, 2008; Yukon Department of Education, 2007). For example, the stated purpose of one communication and information technology (CIT) curriculum is:

> … to focus on how CIT can be used from grades 1 to 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. (Prince Edward Island [PEI] Department of Education, 2008, p. 15)

It is clear that the intent of this curriculum is to use computer technology in natural, everyday contexts to solve problems encountered by students in daily life. Accordingly, technology use is not superfluous but rather an integral tool for teaching the core subjects. The integration of computer technology into instruction, where teachers consciously plan for the use of computers to support student learning, forms the basis of computer-assisted instruction (CAI).

Although computer use has well-documented benefits, computer technology has not been well integrated into primary classroom instruction (Bauer & Kenton, 2005; Becker, 1998; Dwyer, 2007; Judge et al., 2006; Norris, Sullivan, Poirot, & Soloway, 2003; Wozney, Venkatesh, & Abrami, 2006). Few studies examine different methods of integration, the extent of technology integration, or suitability of authorized software programs for teaching reading or writing in mainstream classrooms. Rather, it is more often the case that studies examine total technology integration using software and hardware unavailable in most mainstream classrooms or specific software for intervention studies in special education. The research herein reports on the specific software programs authorized by the provincial and territorial programs of study to instruct or supplement the teaching of reading and writing in the primary grades in Canada.

Technology Integration in Classroom Instruction in the Primary Grades

Bauer and Kenton (2005) defined technology integration as "a reliance on computer technology for regular lesson delivery" (p. 522). They used Hooper and Rieber's (1999) continuum of teachers' use of technology within the classroom from (a) familiarization, (b) utilization, (c) integration, (d) reorientation, to (e) evolution to study why technology is not happening in schools. Bauer and Kenton (2005) found that most teachers are either familiar with computers but do not use them in their teaching practice, or utilize computers as superfluous to the regular curriculum. Non-technologically savvy teachers frequently do not progress past the utilization stage, where "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). Technologically savvy teachers do not move past the first stages of technology integration on the continuum, and few teachers progress to reorientation and evolution, where they use technology as an essential feature of their delivery of core subject content to their students.

Several researchers report concerns with the limited impact of technology in modern classrooms. For example, Norris et al. (2003) concluded, "Although the literature points to the potential for impact, the reality is sobering: to a first-order approximation, the impact of computing technology over the past 25 years on primary and secondary education has been essentially zero" (p.15). Barriers to technology integration in classrooms include lack of teacher knowledge of technology, a low level of comfort with technology, lack of adequate preparation and planning time (e.g., Dwyer, 2007; Wozney et al., 2006), and lack of support from administrators in the allocation of resources (e.g., Dwyer, 2007). Furthermore, teacher and administrator attitudes toward technology use may affect how computers are used. For example, Dwyer (2007) found that teachers and administrators in Australia reported computer use to be inappropriate for primary students and favoured traditional teaching methods, with the result that newer resources were allocated to older students and slower, outdated computers were allocated to primary students. Dwyer also reported that in some cases they practiced their keyboarding skills on cardboard models. Yet, despite reports of significant barriers to meaningful technology integration in teachers' classroom practice, Canadian teachers are expected to follow the technology curricula mandated by their respective provincial and territorial ministries of education, which, in many cases, mandate that technology be integrated into classroom instruction wherever and whenever appropriate.

Canadian curricula are typically developed by subject-area committees comprised of consultants from within each ministry as well as teachers appointed from rural, suburban, and urban school districts. Each committee develops programs of study that outline the knowledge, skills, and attributes that students are to attain in each subject area. These programs are recommended for approval by their respective provincial or territorial ministry. When curricula are approved, they are a legally binding program of study to be followed by all (K–12) teachers within each respective province or territory. There is overlap in curricula because regional provinces and territories confer with one another and thus often include most of the same goals, objectives, and authorized resources including commercial software programs.

How Computer Software Is Used to Teach Reading and Writing Skills

Several recent studies have evaluated the impact of using computer software with students either to enhance or assess reading and writing skills. The multimedia capabilities of modern computers, especially text-to-speech capabilities and graphical representations, support the teaching of phonological awareness skills (Bishop & Santoro, 2006; Fasting & Lyster, 2005; Littleton, Wood, & Chera, 2006) and letter names to young children or struggling readers (Connell & Witt, 2004); assist in building vocabulary skills (Gill, 2007); and improve reading fluency and comprehension (Sorrell, Bell, & McCallum, 2007). Also, software programs have been developed to teach the writing process or to scaffold writing development (Englert, Manalo, & Zhao, 2004; Wade-Stein & Kintsch, 2004). However, these programs appear to be designed for older students. It is encouraging to note that, in contrast to Dwyer's (2007) and Becker's (1998) suggestion that most teachers, especially primary teachers, use computers for drill-and-practice activities, many of the writing programs were open ended and allowed for exploratory learning.

Recently, some of the educational benefits of software for teaching reading and writing in the primary grades have been reported. Macaruso, Hook, and McCabe (2006), and Macaruso and Walker (2008) found that students in kindergarten and elementary classrooms, where considerable amounts of systematic computer-assisted instruction was integrated into their language arts curricula, showed significant gains in reading compared to control groups, particularly for the lowest-achieving students. After 45–60 sessions of computer-assisted instruction over a six-month period, students were found to make gains in their phonological and reading skills. In both studies, Macaruso and his colleagues used software programs that targeted specific skills in sequence, offered feedback to students and teachers, and automatically branched to address remediation activities as required. These results bolster Bishop and Santoro's (2006) requirement that educational software have an evaluative component and provide feedback so that the program can be adapted to meet students' educational needs as well as monitor their progress. Without the evaluative and feedback components, technology use in the classroom likely will not be effective as educational tools.

Even though computer use has been linked to benefits, teachers must be cognizant of the reasons for using technology in the classroom and make informed choices about the specific educational benefits and drawbacks of software programs. Sorrell et al.'s research (2007) highlights the issue of instructional appropriateness of software programs. They found that the use of one assistive program, Kurzweil 3000, actually hindered reading fluency and comprehension among proficient readers and concluded, "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). Similarly, Macaruso and Walker (2008) cautioned, " … while CAI is a key contributor to reading gains in low-performing kindergarteners, other factors play an important role as well" (p. 280). Hence, caution is justified when using computer-assisted instruction, and other instructional approaches are not to be ignored if appropriate for meeting students' educational needs.

Parents and teachers have widely accepted that technology is an important tool for engagement in literacy instruction. More specifically, it is assumed that if students experience difficulties with areas such as word recognition, reading speed, and writing, software programs will provide much needed extra practice. Even though it is known that the type and quality of the programs and students' specific needs affect program effectiveness, the general theory is that educational software programs are beneficial. They complement classroom instruction; provide opportunities for students to practice skills; and enable students to enjoy immediate individual feedback, work independently, and gain a sense of accomplishment. Thus, they are beneficial when integrated with reading and writing instruction in classrooms. Whether the theory holds in practice is of interest to us.

Purpose and Research Questions

Teachers are expected to meaningfully integrate technology use into their daily teaching practices. One way to integrate technology is through the use of authorized commercial software programs for teaching reading and writing. Four specific questions guided our research:

- What authorized software programs are available?
- What are manufacturers' claims about their software programs, and are the claims supported?
- Are reading and writing prerequisite skills for using the software programs?
- Are these software programs appropriate and useful to supplement classroom reading and writing instruction?

METHOD

Data Sources

The authors compiled a list of 47 programs authorized for use by the ministries of education in the provinces and territories of Canada (excluding Quebec and Nunavut, which both predominantly instruct children in languages other than English, and New Brunswick and Newfoundland, which did not provide central listings of authorized software) in February 2008 by searching the online resource catalogues with the search terms reading and writing and contacting the Ministries of Education for each province and territory by email and telephone over a three-month period from December 2007 through February 2008. The list of 47 programs represented the total, but some of the programs on the list were used in multiple provinces and territories. For example, Kids Works Deluxe (1996) was in listed in Alberta, British Columbia, Manitoba, and the Northwest Territories, whereas Clicker 5 (2006) was listed only by the province of Ontario. Of the 47 programs, only 13 (28%) were available for purchase and therefore for analysis in this research. Twelve of the 13 programs were published in the United States, and one was published in Canada (SMART Ideas, 2005). Twenty-four (51%) of the programs were either outdated or no longer distributed by their publishers (some appeared to be available from Europe but were on unsuccessful backorder for months), or older but very similar versions of the same program. The remaining 10 programs (21%) were deemed for special needs students only or "other" software that supported paper resources such as CD-ROMs accompanying spelling program manuals. The authors procured copies of all available authorized reading and writing software programs for in-depth analysis.

Unit of Analysis

Most programs had accompanying user manuals listing technical specifications and explaining how to use the program. A few had supplementary resources such as sample activities for teachers; however, most of the programs were not designed exclusively for school use and did not include supplementary teacher resources. The authors analysed separately as a unit and evaluated each available program and accompanying support material.

Development of the Evaluation Instrument

After considering several evaluation models and instruments, the authors judged the Bishop and Santoro (2005, 2006) instrument to be the most suitable for evaluating early reading and writing software based on the following criteria: (a) the instrument is comprehensive, (b) the instrument is based on research in each of the evaluation areas (interface design, content, and instructional design), (c) the instrument underwent several iterations of pilot testing (Bishop & Edwards, 2003), (d) the instrument provided high interrater reliability (93%), and (e) the instrument corresponded with the purpose of the research reported herein—to evaluate the suitability for instructional use of several commercially available software programs for young children. However, the Bishop and Santoro (2005) instrument was designed for use with a specific type of instructional software (early phonological awareness) and thus had to be modified, as explained in a subsequent section under the content category.

The researchers deemed other models and instruments to be less suitable. For example, McVee and Dickson's (2002) rubric was based on their ideas of useful software traits and was not pilot tested, and most models and instruments were used to evaluate gains in student achievement after specific software use (Englert et al., 2004; Fasting & Lyster, 2005; Lefever-Davis & Pearman, 2005; Littleton et al., 2006; Mostow, Aist, Burkhead, Corbett, Cuneo, Eitelman et al., 2003; Wade-Stein & Kintsch, 2004) rather than program design or suitability.

The researchers analysed each available authorized software program for interface design, content, and instructional design using an adapted version of the Bishop and Santoro (2005) Early Reading Software Evaluation Form. The adapted evaluation addresses three main areas of interest: the quality of software design, skills taught through use, and instructional soundness of the software program to deliver the targeted skills or knowledge. Specifically:

1. Interface design criteria and indicators examine the overall design of the software program, including whether the program's visual and auditory media are aesthetically pleasing to young children, how instructions

are relayed to children, how errors are handled with the children, and whether the software program is interactive with and responsive to children's engagement.

2. Content criteria and indicators examine what is actually taught or addressed by the program. To accommodate the wider scope of our research, the original Bishop and Santoro (2005) content criteria on phonological awareness were modified to accommodate our interest in whether or not the program(s) met the stated educational objectives for reading and writing claimed by the software manufacturers and publishers.
3. Instructional design criteria and indicators examine the educational soundness of software programs. They include how systematically the program content is delivered, whether the program teaches through the use of explicit examples and practice, how the program supports users who experience difficulties, and how the program records and evaluates student responses and provides educational recommendations based on the evaluation. The Motivation subcategory was removed because the indicators were deemed to be subjective.

For a complete list of the categories and indicators used, see the Appendix on pages 214–216.

Evaluation Procedure

The researchers methodically followed the Bishop and Santoro (2006) guidelines for assessing software features in two related phases. Phase one (activities 1 to 7 follow a program as a child would use it to assess ease of use by children, level of engagement, and appropriateness) and phase two (activities 8 to 10 follow a program as a teacher might use it to assess for appropriateness and suitability in meeting educational goals).

Phase One

1. The researchers ran each available software program to experience first-hand how children might use the program. We did not read the manual before launching the title sequence to play the program.
2. The researchers stopped and launched the program again, bypassing the title screen if possible to study the consequences. For example, many children bypass the beginning sequences of computer programs, and thus it is important to see whether they bypass necessary instructions to make the program useable.
3. The researchers checked the program to establish whether it requested a login. (Bishop and Santoro [2006] noted that if students are not prompted to log in, the program does not track student performance and growth over time.)
4. The researchers assessed the design of the interface (font, graphics, and music) for suitability, consistency, organization, and flexibility according to the criteria on the evaluation instrument.

5. After assessing interface design, the researchers considered the quality of instructions at first use and throughout the activities. Given that young children are meant to use these programs, it was important to check whether program instructions were clear, repeatable, and presentable both orally and visually.
6. Next, the researchers made deliberate navigation and content errors to simulate errors children might make to assess how the program handled unexpected input. We made an attempt to navigate in unexpected ways through the program as well as to answer questions incorrectly, omit some questions, and repeat the same answers to multiple questions. (These types of errors were done deliberately, as advised by Bishop and Santoro, to determine whether the program will eventually demonstrate how to complete the function or activity correctly, much as a teacher would do in a traditional instructional setting.)
7. The researchers assessed the level of interactivity by noting how many times and for what duration learning activities were interrupted for the presentation of instructions or animated clips.

Phase Two

8. After assessing the software program was assessed from the perspective of how a child might experience it, the researchers reconsidered the program from the perspective of a teacher. First, we read the teacher's manual and accompanying material, and then we examined the educational claims.
9. The researchers ran through the program again, testing each part to determine how the program actually addressed the stated skills (i.e., whether the program provided for sequential skill development from easier to more complex; modelled skills before students performed them; and gave focused, timely, and specific feedback when required). In addition, we closed the program during tasks to check whether the program restarted or continued from the point at which it closed by saving student work to that point.
10. We assessed the types of feedback provided (i.e., what feedback was provided for teachers to make educational decisions, and how was remediation, extra practice, or advancement to the next activity designed and managed). This last step is critical in determining the educational value of a program. For example, if a student is having difficulty with a skill while using the program, it is important to know whether or not the program automatically demonstrates the skill again and provides additional practice before moving on to the next skill.
11. The researchers analysed each software program completely at least five times and more often depending on length. We converted the indicators for each category (interface design, content, and instructional design) into a Likert scale from 1 (strongly disagree) to 5 (strongly agree) and tallied the scores for each. The first author scored all software programs, and an independent reviewer (who also viewed the programs) established

Table 1: Mean Percentage of Total Possible Score by Program Type, Title, and Evaluation Category

		Evaluation Category		
Type	Title	Interface Design	Content	Instructional Design
Drill and Practice		77	74	64
	A to Zap! (1998)	67	64	51
	Bailey's Book House (1995)	77	86	71
	Reader Rabbit 1 Deluxe (1994)	88	73	69
Concept Mapping		78	81	67
	Draft: Builder Solo (2005)	81	76	89
	Inspiration 8 (2006)	75	87	60
	Kidspiration 2 (2005)	80	80	61
	SMART Ideas Concept-Mapping Software (2005)	74	80	59
Word Processing		82	73	81
	Clicker 5 (2006)	82	68	72
	Write: Outloud Solo (2005)	81	78	89
Desktop Publishing		74	73	56
	Easy Book Deluxe (1998)	68	65	47
	Kid Works Deluxe (1996)	83	73	54
	Storybook Weaver Deluxe (1996)	72	70	46
	Ultimate Writing and Creativity Centre (1997)	71	83	75

interrater reliability by a match-mismatch procedure, which the second author confirmed. Interrater reliability was calculated at 89%. All differences were resolved for the findings reported.

FINDINGS AND DISCUSSION

Types of Available, Authorized Software

The researchers separated the authorized software into four broad categories. Reading and writing programs for classroom use formed the two main categories. We excluded the remaining two from our analysis because they were either designed for use with specific groups of students outside of regular classrooms or supplemental software designed to support resources other than specifically reading and writing, such as listening, speaking, or spelling programs.

Reading programs comprise 28% of the listed software and are divided into two subcategories: drill-and-practice games (17%) and talking books (11%).

Drill-and-practice games are designed around game interfaces to encourage player motivation through the use of sounds, pictures, and bright colours. In addition, children practice or complete drills on one or more skills in a question/answer format, in which answers are typically either right or wrong. Talking books software includes a story and accompanying activities or games around the story theme. The computer reads the story and children read along from the computer screen. Authorized talking books software titles were unavailable for purchase and thus excluded.

Writing programs comprise 51% of those listed and are divided into three categories: concept-mapping software (15%), word-processing software (19%), and desktop publishing software (17%). Software designed for making concept maps allows students to use graphics, text, sound, and even animation to represent ideas in pictorial format, with lines and symbols to show the relationship between and among ideas. Word-processing and desktop publishing programs are similar in function and are used to compose and edit text on the computer with or without graphics, or in different layouts, and then print the results. However, desktop publishing software also focuses on graphic and text layouts and frequently uses a theme (typically a storybook), and students write and publish stories to share. Of the 13 remaining programs, the drill-and-practice games and desktop publishing programs are most dated (1994–1998), and the concept-mapping software and word-processing software are newer and were published in either 2005 or 2006.

Using the adapted Bishop and Santoro (2005) evaluation instrument as the template for assessing the programs, the researchers determined the percentages for each category for each program by assigning a Likert scale with values from 1 (strongly disagree) to 5 (strongly agree) to each indicator, then dividing the sum of the scores for each category by the total possible score to obtain a percentage score. A score in the range of 85–100% (corresponds with the top 25% of possible scores) met or exceeded expected standards, 61–84% adequately met expected standards, and 20–60% (corresponds with the lower 50% of possible scores) did not meet expected standards. Table 1 (page 205) presents the percentage means for each program type, with totals in bold for each evaluation criterion, followed by the totals for each specific program within the program type.

The mean total percentages for interface design across all software program types ranged from 74 to 82. According to the Bishop and Santoro (2005) instrument and the percentage guidelines developed for this research, only Reader Rabbit 1 Deluxe (1994) exceeded expectations, and the rest adequately met expected standards. The highest mean percentage calculated on interface design for an individual program was 88% (Reader Rabbit 1 Deluxe), published in 1994, and the lowest was 67% (A to Zap!), published in 1998, with a range of 21 percentage points. Improvements on interface design were not evident on the basis of how recently the program was published.

The authors used the manufacturers' educational claims to determine the content score by comparing claims against the programs' functions. Of all of the program types, concept-mapping programs scored highest on content (total mean of 81%). Drill-and-practice games scored second but were off by seven percentage points. Generally, the activities provided supported most manufacturers' claims. However, few of the writing programs claimed to teach writing skills such as how to structure or add detail to writing. Although most individual programs scored adequately on content, two programs exceeded expectations: Bailey's Book House, published in 1995, and Inspiration 8, published in 2006. The highest individual mean percentage score calculated for content was 87% (Inspiration 8), and the lowest was 64% (A to Zap!), with a range of 23 percentage points. The remaining ten programs included five published in the 1990s and five in the mid-2000s, suggesting that date of publication did not have a corresponding improvement in content.

Instructional design was rated the lowest overall of the three evaluation categories. This finding is consistent with many of the programs evaluated and reported on by Bishop and Santoro (2006). Of the 13 programs analysed, two exceeded expectations: Draft: Builder Solo (2005) and Write: Outloud Solo (2005); five met expectations: Bailey's Book House (1995), Clicker 5 (2006), Kidspiration 2 (2005), Reader Rabbit 1 Deluxe (1994), and Ultimate Writing and Creativity Centre (1997); and six did not meet expectations: A to Zap!, (1998), Easy Book Deluxe (1998), Inspiration 8 (2006), Kid Works Deluxe (1996), SMART Ideas 5.0 (2005) and Storybook Weaver Deluxe (1996). Two programs were rated the highest (89%): Draft: Builder Solo (2005) and Write: Outloud Solo (2005), and the lowest, Storybook Weaver Deluxe (1996), scored 46%, for a range of 43 percentage points. Most programs scored poorly on instructional design because they did not meet the requirements set by Bishop and Santoro (2006): They did not track student progress, perform error analysis, automatically adapt to provide extra remediation when required, or report results to enable teachers to make informed educational decisions.

According to our findings, instructional design ratings are not date sensitive. The two programs rated the highest were published in 2005, and four other programs also published in either 2005 or 2006 adequately met expectations, but so did three that were published more than a decade ago. Thus, date of publication alone is not a sufficient criterion on which to judge a program.

Manufacturers' Claims

Manufacturers' claims of educational benefits vary for each type of program analysed. Three drill-and-practice games for young children made a general and primary claim to improve letter and word recognition using picture and sound referents, and these were supported directly with activities in each program. Secondary claims made by the manufacturers of these games included teaching:

- Alphabet skills (such as matching lower and uppercase letters and demonstrating understanding knowledge of alphabetical order)
- Word reading skills (such as understanding that letters combine to make words, recognizing the beginning sounds of words, spelling pattern recognition, and rhyming words)
- Vocabulary skills (such as the acquisition and use of new words)

- Reading and thinking skills (such as increased reading fluency, visual discrimination, and critical thinking)

Letter and word recognition skills were given precedence over the secondary skills in each activity. For the target age levels specified by the manufacturers of the drill-and-practice games, the focus on improving letter and word recognition skills was appropriate. However, these skills are practiced rather than taught, so students must have prior knowledge of the skills in order to practice them. In addition, programs that list many different supplemental skills, such as A to Zap! (1998), do not address any one skill in an in-depth manner.

The four concept-mapping software titles identified in Table 1 functioned similarly and have comparable manufacturers' claims: clarifying thinking, enhancing understanding, promoting visual learning, and increasing concept retention. These programs also claimed to assist students with the writing process, especially organizational and prewriting skills, as well as the addition of relevant supporting details to enhance their written language. Supplementary claims made by these programs included teaching children how to edit spelling at a developmentally appropriate level, to write in a paragraph format, and to write coherently.

The concept-mapping software programs were judged to be noninstructional on the basis of the criteria specified by Bishop and Santoro (2006). Students use concept maps to organize ideas and plan drafts for writing, but these skills are not specifically taught. The software is open ended unless teachers use templates to teach writing skills, such as paragraph format. Students must use thinking and writing skills to create concept maps, but these skills are neither directly taught nor modelled by the program, so the manufacturers' claims are supported only partially. One concept-mapping program, Draft: Builder Solo (2005) claimed to help students develop a purpose for reading. However, a sample assignment was to create a character map for a favourite book. The assignment required reading or remembering a favourite book, but the program did not set a clear purpose for reading. Manufacturers of concept-mapping software further claim their products promote organization and writing skills, yet we found that proficiency of these skills is expected by but not taught through use of the programs.

The researchers analysed two text-to-speech word-processing programs: Write: Outloud Solo (2005) and Clicker 5 (2006). Manufacturers' claims on the former include that the program teaches students to write in paragraph format using indentation, topic sentences, main ideas, introductions, and conclusions. In addition, manufacturers claim that students learn to edit grammar and spelling at developmentally appropriate levels and use reading skills and strategies to understand a variety of informational texts. The other text-to-speech software program, Clicker 5 (2006), did not list any educational claims in the user manual but claimed it to be "a writing support and multimedia tool for children of all abilities and people with special needs" (Clicker 5 User Guide for Windows and Mac, 2005, p. 7). The text-to-speech functions of both programs may assist with writing and spelling strategies by reading back students' writing or increase reading fluency by reading aloud with students, but the computerized voices were typically in monotone without stress or intonation and with breaks between words that did not match normal speech. In addition, the computerized voices often read too quickly to be followed by children reading aloud; in fact, it was a challenge for us. We judged the word-processing programs examined to be noninstructional tools, despite manufacturers' claims.

Unlike the concept-mapping and word-processing programs we analysed, the desktop publishing software programs were outdated by at least a decade (see Table 1). Similar to word processing, desktop publishing programs are designed for composing text but mainly to format text and pictures into different layouts. A common claim made by manufacturers of desktop publishing software is that students demonstrate and use their understanding of the writing process (prewriting, writing, revising, editing, and publishing). In practice, most of these programs, with the exception of Ultimate Writing and Creativity Center (1997), focus on writing and publishing, ignore prewriting and revision, and perform automatic editing functions for students rather than guiding students to edit their grammar and spelling independently. For example, the program will check spelling for the students and may indicate incorrect words by underlining them or making a sound. Although students have the option to accept or ignore the proposed change in spelling, no further help is offered should the proposed word not match the intended word. The claim to help in all areas of the writing process could not be substantiated.

Reading and Writing as Prerequisites for Software Programs

The drill-and-practice games were designed for use in preschool to first grade. These were designed as games and make extensive use of computer-generated speech for reading directions and responding to questions. Unfortunately, many of these games either ignore as incorrect any answers that do not match expected responses, or tolerate and gradually eliminate incorrect responses merely by repeated clicking. Reading is not a prerequisite for playing the games, because eventually correct answers are achieved either through chance or elimination. For example, in Reader Rabbit 1 (1994), children match vowels to pictures containing the vowel sound during a memory game, and incorrect responses are ignored by the program. In Bailey's Book House (1995), when children are creating rhymes, incorrect responses are removed gradually so that, in the end, only the expected response remains. Writing is not a prerequisite for any of these games, even though they provide opportunities for students to practice phonics skills. Inability to practice skills in a natural context by decoding words through reading text rather than by playing a game could hinder the transfer of skills outside of the game context and may mean students do not retain the skills when needed for authentic reading.

Writing is a prerequisite for concept-mapping software. Students write text for labels and concepts on the map and are then encouraged to write and expand their ideas using the outline functions. The stated purpose is to help students to plan and organize their writing. Reading is not a prerequisite for creating concept maps, but students typically read their own writing, the instructions, and other print information.

As the function of word-processing and desktop publishing software is to facilitate writing and the writing process, writing is a prerequisite for using the software programs. These programs are on the whole noninstructional; they provide practice of writing skills but do not directly teach writing or the writing process. One notable exception is Ultimate Writing and Creativity Center (1997), which provides explanations and hints for each part of the writing process. Reading is not a prerequisite skill for either word-processing or the desktop publishing software programs. However, many programs in both categories read back student writing, which may indirectly improve their reading fluency as they read along while the program highlights the words. Clicker 5 (2006), Kids Works Deluxe (1996), and several other programs record students' voices for inclusion with their writing. Improving student reading fluency is not a primary function of the software programs, and assessment functions for reading fluency are not included.

Software Programs as Supplements to Reading and Writing Instruction

Of the 13 software programs analysed, none were judged directly instructional according to the criteria specified by Bishop and Santoro (2006). Few demonstrate reading and writing skills, and most neither monitor student progress nor offer corrective feedback. We speculate that the programs do not monitor or provide corrective feedback because most of these are designed for commercial sale to parents of small children, and schools are a secondary audience. On the other hand, each program is age appropriate and could be integrated easily into language arts lessons to enhance reading and writing skills. However, none of these programs could be used to teach reading or writing skills for the reasons outlined.

Each of the drill-and-practice games was designed for use by preschool, kindergarten, or first grade students for learning letter recognition, letter sounds, and word recognition skills. The programs were developed for home and school use and are appropriate for supporting early reading development. The dual home and school use results in inherent flaws that hamper the educational use of drill-and-practice games in classrooms. Flaws include disregarded incorrect choices and uncontrolled continued clicking, which allow these games to be played without reading or using the targeted skills until the only remaining option is the correct one. In addition, replay is limited because programming restrictions prohibit the addition of new content and the adaptation of existing content to suit individual needs for remediation based on error analysis.

We judged the writing programs across concept-mapping software, word-processing software, and desktop publishing software to be noninstructional. Unless a teacher specifies the format and content of the finished work or uses a template, the students' written products are open ended. Each of the programs in the writing category assisted with aspects of the writing process, but none assisted with the entire process. For example, concept-mapping software was designed to help with the prewriting portion of the writing process but had limited tools for drafting, revising, and editing. Desktop publishing software was designed to assist in publishing documents and contained drafting capabilities but had limited editing and revising tools and often excluded prewriting.

On balance, the programs appeared to be suitable tools to facilitate writing for students already proficient for their grade.

CONCLUSIONS AND RECOMMENDATIONS

Several obstacles hinder technology integration in the classroom. Despite the reported benefits by researchers (such as Macaruso et al., 2006; Macaruso & Walker, 2008; Norris et al., 2003; Page, 2002; Sorrell et al., 2007; and others) of the appropriate use of facilitatory software as tools to support learning, some of the evaluated programs do not support skill development in a natural context. Moreover, the manner in which programs are used is crucial. Desktop publishing software claimed to support the writing process, yet if teachers do not use computers for all aspects of the writing process, such as researching and organizing ideas, preparing first drafts, or editing and revising multiple drafts, it follows that the software is not being used to support all aspects of the writing process. Partial use of programs appears inconsistent with provincial and territorial technology integration requirements, such as the PEI Department of Education's (2008) requirement that students use technology as a problem-solving tool in the core subjects. A second obstacle is that many software programs were designed on what is now considered to be decade-old technology and ideas. Many of these programs are therefore not able to perform at a level consistent with modern programs, as evidenced by the quality of the media (voice, graphics, and design interface) and program functions (such as tracking and reporting student performance and the flexibility of activities to allow for individual instruction).

Our results confirm that technology integration is not a priority in the primary schools in Canada, particularly as many of the authorized materials on current lists are outdated and unavailable. Up-to-date software programs are widely available commercially, but they have not found their way onto the authorized software lists in many cases. For instance, some ministries of education sites have not been updated for close to a decade, and the turnaround time for new curriculum development in a specific area can be on a 10–13 year cycle. Given that available technology inexorably improves in speed, power, availability, and facility, research is needed on how technology can best be used in pedagogically sound ways to support programs of study. Our results further confirm that interface design, content, and instructional design do not necessarily improve with more recent publication. Hence, research is needed on how to improve these three criteria in educational software programs.

Based on the results of our study, two serious problems are clear. First, the outdated lists contain programs that are not tools for the meaningful integration of computer technology into instruction. Second, the majority of the programs are noninstructional; they do not track student progress, provide feedback, or adapt to suit student needs, thereby limiting their usefulness as educational tools.

Manufacturers' claims are often sweeping, and although they use educational vocabulary, claims of educational gains are not supported by evidence from experimental trials and systematic analyses. Trials may have occurred, but they are not reported either in the literature accompanying the programs or elsewhere. Consequently, teachers, schools, and school boards face yet another shortcoming in the amount of reliable and valid evidence to determine whether or not programs are pedagogically appropriate or effective. We expect some teachers and students are well ahead of the commercial authorized resources and use technology as a tool for teaching and learning in impressive ways. Nonetheless, the question remains whether the authorized programs represent the actual uses of technology in classrooms or whether other programs are being used and how. Teachers' ideas on the best software programs and how they integrate them in pedagogically sound ways in their teaching of reading and writing are the next steps in our research.

Contributors

Meridith Lovell is a primary school teacher and provisional PhD candidate in language and literacy at the University of Alberta. (Address: Canadian Centre for Research on Literacy, 653 Education South, University of Alberta, Edmonton, Alberta, Canada T6G 2G5; Phone: +1. 780.492.5090; E-mail: malovell@ualberta.ca)

Linda Phillips is a professor and director of the Canadian Centre for Research on Literacy at the University of Alberta. (Address: Canadian Centre for Research on Literacy, 653 Education South, University of Alberta, Edmonton, Alberta, Canada T6G 2G5; Phone: +1. 780.492.5090; E-mail: linda.phillips@ualberta.ca)

References

A to Zap! [Computer Software] (1998). Elgin, IL: Sunburst Technology.

Alberta Learning. (2000). *Information and communication technology.* Edmonton, AB: Author. Retrieved on February 1, 2008, from http://education.alberta.ca/media/453069/pofs.pdf

Bailey's Book House [Computer Software] (1995). San Francisco: Riverdeep Interactive Learning Ltd.

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., & Edwards, L. E. (April, 2003). *Technology applications for children with early literacy difficulties: A framework for review and evaluation of popular beginning reading computer software.* Paper presented at the annual meeting of the American Educational Research Association, Chicago, IL. Retrieved September 21, 2009, from http://www.lehigh.edu/~mjba/TABR/pdf/AERApaper_041903.pdf

Bishop, M. J., & Santoro, L. E. (2005). *Early reading software evaluation form.*

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

Bethlehem, PA: Authors. Retrieved on February 1, 2008, from http://www.lehigh.edu/~mjba/TABR/pdf/ReadingSWInstrument _090205.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.

Clicker 5 [Computer Software] (2006). Redmond, WA: Crick Software, Incorporated.

Clicker 5: *User Guide for Windows and Mac* (2005). Redmond, WA: Crick Software, Incorporated.

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.

Draft: Builder Solo [Computer Software] (2005). Volo, IL: Don Johnson Incorporated.

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.

Easy Book Deluxe [Computer Software] (1998). Elgin, IL: Sunburst Technology.

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.

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

Hooper, S., & Rieber, L. P. (1999). Teaching, instruction, and technology. In A. C. Ornstein, & L. S. Behar-Horenstein (Eds.), *Contemporary Issues in Curriculum* (pp. 252–264). Boston: Allyn and Bacon.

Inspiration 8 [Computer Software] (2006). Beaverton, OR: Inspiration Software Company.

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.

Kid Works Deluxe [Computer Software] (1996). Overland Park, KS: Davidson and Associates Incorporated.

Kidspiration 2 [Computer Software] (2005). Beaverton, OR: Inspiration Software Company.

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.

Macaruso, P., Hook, P. E., & McCabe, R. (2006). The efficacy of computer-based supplementary phonics programs for advancing reading skills in at-risk elementary students. *Journal of Research on Reading, 29*(2), 162–172.

Macaruso, P., & Walker, A. (2008). The efficacy of computer-assisted instruction for advancing literacy skills in kindergarten children. *Reading Psychology, 29*, 266–287.

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.

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 (2008). *Elementary program of studies and authorized materials 2008–2009.* PEI: Author. Retrieved June 15, 2009, from http://www.gov.pe.ca/photos/original/ed_elemps_0809.pdf.

Reader Rabbit 1 Deluxe [Computer Software] (1994). Fremont, CA: The Learning Company.

SMART Ideas ® Concept-Mapping Software [Computer Software] (2005). Calgary, AB: SMART Technologies.

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.

Storybook Weaver Deluxe [Computer Software] (1996). Minneapolis, MN: SoftKey Multimedia Incorporated.

Ultimate Writing and Creativity Centre [Computer Software] (1997). San Francisco: Riverdeep Interactive Learning Ltd.

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.

Write: Outloud Solo [Computer Software] (2005). Volo, IL: Don Johnson Incorporated.

Yukon Department of Education. (2007). *Yukon K–7 ICT curriculum: Information and communication technology.* Whitehorse, YT: Author. Retrieved February 1, 2008, from http://dl1.yukoncollege.yk.ca/ K7ICT/

APPENDIX: EVALUATION CATEGORIES AND INDICATORS

(Adapted from Bishop & Santoro, 2005)

Interface Design: To what extent is the software:

1. Aesthetically pleasing? Does the program's interface use media (text, graphics, animations, video, sound) in ways that enhance the experience?
   a. The media used is high quality.
   b. Screens are laid out in well-organized ways (rather than haphazard placement of objects).
   c. Screens are neither overly stimulating nor boring.
   d. The "look and feel" of this program is likely to be pleasing to the learner.
   e. Media are used to create themes/metaphors that relate to the content and help create meaning.
   f. Learner is able to modify the interface according to individual preferences.

2. Supportive operationally? Will the prereading learner be able to use the program with little help from adults?

Direct Support:

   a. All operational instructions are supplied auditorially within the program.
   b. Operational instructions can be reviewed, as necessary.
   c. Instructions supplied within the program will be helpful to the intended audience.
   d. The interface responds with prompt and informative invalid action messages when appropriate.
   e. After repeated invalid actions, the interface shows the learner how to correctly operate the function.

Indirect Support:

   a. The interface takes advantage of what learners already know.
   b. Learners don't have to search for commonly used functions.
   c. Program functions are placed in equivalent, if not identical, locations on screens.
   d. Things on the screen are what they appear to be and function as expected.

3. Interactive? Is the learner the primary driving force behind what happens in the program?
   a. The learner rarely sits passively watching as the program does things.
   b. Interactions are frequent.
   c. The learner interacts directly with screen objects.
   d. Interactions with screen objects are as nearly like their real-world referents as possible.
   e. Learner interactions make a substantive difference in what the program is doing.

Instructional Design: To what extent is the software:

1. Systematic? Is the instruction comprised of cycles that progress hierarchically through increasingly difficult blocks of content and skill sets?
   a. The program gains learners' attention at the beginning of each instructional cycle.
   b. Learners are reminded of prerequisite knowledge at the beginning of each instructional cycle.
   c. The program informs learners of objectives at the beginning of each instructional cycle.
   d. The program offers multiple examples of a target skill (including use of pseudowords).
   e. The program supplies adequate opportunities for learners to practice newly learned skills.
   f. Learners must demonstrate mastery of previously introduced skills before moving on to new skills.
   g. The program supplies larger conceptual anchors for retention and retrieval (transfer of knowledge).

2. Instructionally supportive? Does the program supply appropriate levels of content support to enhance learning?
   a. The program makes content support available precisely when the learner needs it.
   b. The content support provided is helpful, but not so prescriptive that it short-circuits learning.
   c. The program uses informative, instantaneous feedback messages to support content learning.
   d. The program branches automatically to accommodate learner's remediation needs.
   e. The relevance of learning activities is made clear to the learner.

3. Assessing? Does the program evaluate learner progress and help direct learning goals?
   a. The program saves learners' work
   b. The program supplies progress summaries.
   c. The program graphs or charts learner performance in an easily interpreted way.
   d. The program interprets learner performance and makes recommendations for how to proceed.
   e. The program includes an administrative function that tracks all learners working with it.