Formative Evaluation for Implementation: Evaluating Educational Technology Applications and Lessons

TIM WESTON

ABSTRACT

Educational technology applications and lessons, such as learning modules, activities, and user-authored systems, face substantial implementation challenges in public schools and post-secondary institutions. The design of technology impacts the willingness of teachers to implement new technology; evaluators examining implementation obstacles can help improve design. In this article, both summative and formative implementation evaluation are discussed in non-technology contexts and compared with current practices with educational technology. Formative evaluation has traditionally focused on making applications and lessons “work”; however, many implementation problems are caused by non-technical/interface concerns and related to the compatibility of the technology with its surroundings. An overview of research on implementation of applications and lessons is provided and linked to a model of formative evaluation for implementation in this area. Technical, curricular/instructional, and practical areas of implementation are examined. Common implementation problems and typical design recommendations are presented. The use of the model is discussed in other areas, such as medicine and assistive technology.

INTRODUCTION

Educational technology applications and lessons, such as user-authored software systems, simulations, and learning modules, face substantial difficulties being adopted and implemented in public schools and post-secondary institutions. The exponential growth in ownership of personal computers (the most common medium for applications) hides the fact that computers are under-utilized in schools (Becker, 1998; Cuban, 2002; National Center for Educational Statistics, 2000). Reasons for slow adoption and implementation include limited or problematic access to resources (e.g., appropriate hardware), cost, and poor technical support in schools, as
well as reluctance on the part of teachers to change tried-and-true instructional practices and take risks with new technology (Ertmer, 1999). Given these formidable barriers to technology integration, good design may tip the balance in the decision to use and implement an application. Difficult to use products are commonly ignored by teachers, while simple, compatible and effective designs have a better chance of classroom integration.

Formative evaluation, one way of improving the design of learning technology, has typically concentrated on design modifications for the usability and curricular content of computer applications. Evaluators help debug software and gather feedback to make applications work better. With formal lessons, such as integrated learning systems, distance education, and web modules, evaluators help coordinate expert review and conduct field studies in the classroom with learning products. Both usability and curricular evaluation make valuable contributions to well-designed, usable learning products, but data collection tends to be limited to answering the question: Does the application or lesson work? The discussion of formative evaluation presented in this article examines a question not typically covered in many evaluation efforts or related literature, namely: Does the application or lesson fit into the context of schools? Formative evaluation of implementation examines the technical, curricular, and practical factors that inhibit the implementation and compatibility of an application. Evaluators analyzing classroom use can help developers more closely match designs to conditions found in schools, thus making it easier for instructors to implement new applications and lessons.

EDUCATIONAL TECHNOLOGY APPLICATIONS AND LESSONS

Before examining the formative evaluation of contextual obstacles to use, it helps to understand the forms and purposes of present day educational technology. The educational technology discussed in this article includes applications, such as computer software (and applications delivered by the Internet) and computer delivered lessons used in schools, colleges, and universities. In general, these technologies provide students easier access to information, help students develop knowledge and skills, and link people in different locations (Brown, Bransford, & Cocking, 2000; Knapp & Glenn, 1996). In classrooms, applications and lessons support lectures, discussion, individual and group learning, and other class activities. Outside the classroom and in computer labs, the same technology is used to support independent study, homework, tutoring, skills development, and communication, or to aid in class-related research. The Internet and its applications are also used in distance education, both as a means of delivery and communication, and as an aid to learning and instruction. Designers of educational technology commonly take advantage of existing non-educational technologies (mostly hardware and systems) as vehicles for educational applications. The use of radio, television, and personal computers all started out in non-educational settings but eventually found a place in schools. In recent years, developers and instructors have adapted the use of email and the World Wide Web for instructional purposes. Presently, innovations in learning technology come from the areas of artificial intelligence, voice recognition, virtual reality, and other emerging technologies.

Three primary types of learning technology applications are used by instructors in secondary and post-secondary institutions: formalized lessons, activities, and user-authored systems. While instructors use technology for other purposes both inside and outside of classrooms, these three categories of educational technology applications and lessons are commonly developed by instructional designers, content specialists, engineers, and scientists with the goal of sustained use by teachers and students.
Formative Evaluation for Implementation

Formalized Lessons

Pre-packaged lessons, programs, modules, and curricula are products developed by publishers, universities, academic laboratories, or other organizations. In the past, these were often limited to distance education and were delivered by radio, television, or as “programmed instruction” on computers (Cuban, 1986). In recent years, the web has made it possible to deliver lessons and modules through the Internet, not only for distance education, but also as an adjunct to classroom instruction. Modules include not only text and pictures, but also exercises, electronic bulletin boards for communication, and other features that take advantage of the online format and interactive capabilities of computers (Weston & Barker, 2001). Integrated Learning Systems (ILS), another type of formalized lessons, are “comprehensive software systems” that deliver instruction and include management systems for teachers (Mills & Ragan, 2000).

Activities

Hands-on applications, such as computer-based exercises, simulations, and games, are often offered as stand-alone activities, either through the Internet or as software packages. Most of these activities are highly interactive and provide immediate feedback to users based on their input. They differ from formalized lessons in that most do not have extensive text, formalized learning objectives, or sequenced lessons.

User-authored Systems

User-authoring systems (Locatis & Al-Nuaim, 1999), such as presentation software, authored simulations, and authored assessments, allow instructors to use “on-screen tools (menus, prompts, icons) that let users enter text, compose graphics, and prescribe branching…” (p. 66). In an instructional context, instead of formal content being written by developers, these systems allow teachers to design their own lesson content with forms and templates, select different functions of an application based on need, and choose content from reference tools. Because authoring is in the hands of the user, the eventual instructional product experienced by students varies widely. The instructional design for user-authored systems concentrates on creating a usable “environment” where instructors are guided to produce lessons.

Classroom Implementation

Formalized lessons, activities, and user-authored systems are all designed for instruction and learning, but designers do not always anticipate the way in which applications and lessons fit—or fail to fit—into the context of schools. Like many other products, educational technology applications and lessons face challenges in implementation. In schools, the benefits of an application, real or perceived by teachers, may be overshadowed by hesitancy to change existing practices. Moreover, applications and lessons function in complex organizational environments where preexisting motivational, organizational, and cultural factors may work against their adoption and sustained use. Fortunately, many educational applications and lessons are highly modifiable and open to alterations in design that address these implementation problems.
LINKING FORMATIVE AND IMPLEMENTATION EVALUATION

Formative evaluation of educational technology applications and lessons reflect broader concerns about implementation found in other areas. Plans made during the early stages of policy-making or program design rarely translate into the reality envisioned by policy makers and developers. Initiatives in prevention, health, mental health, recidivism, and a host of other programs have commonly met with implementation problems. When program administrators, and the intended beneficiaries of programs, implement and respond to interventions, programs change in unanticipated ways because of political considerations, capacity limitations and incompatibility between a program’s design and the entrenched practices of individuals and groups. Almost 30 years ago, Pressman and Wildavsky (1974) described the breakdown between policy and outcomes in a case study of a large federally funded minority assistance program. The program failed to create jobs because of logistical obstacles, politics, and a lack of understanding of how program initiatives fit into local organizational structures and practices. Wildavsky and Pressman pointed out that programs must adapt to their surroundings in order to survive, and recommended that policy makers take the realities of local bureaucracies, conflicting motivations, and other contextual conditions into account when planning programs. A decade later, McLaughlin (1987) summarized two generations of policy implementation studies, illustrating how “implementation dominates outcomes” in a wide range of areas. McLaughlin added that policies succeed or fail not only because of organizational and structural factors, but because of individuals, and that those responsible for implementing programs “do not always do as told,” responding instead in “idiosyncratic, frustratingly unpredictable, if not downright resistant ways” to the efforts of program administrators (p. 172). Because of the unanticipated responses of stakeholders, both in schools and in other areas, policy implementation involves extensive bargaining among stakeholders, iteration between policy actions and responses, and a thorough understanding of the context where the policy must function.

Evaluators examining implementation processes take on summative and formative roles. McLaughlin (1987) reminded evaluators that any generalization made from observed outcomes must be linked to the details of the program’s implementation. Patton (1978) advocated framing evaluation questions “in the context of implementation,” calling for evaluators to examine the substance and everyday workings of programs that lead to observed outcomes. Implementation evaluation describes, first, whether the program reaches implementation (is there a program?), and if so, what it looks like in practice. Describing and analyzing variations in implementation across sites, program strengths and weaknesses, as well as the responses of individuals and groups to program initiatives (and how the program adapted to these local conditions), all are part of what an evaluator does when examining implementation processes.

The same information about the context of implementation that is used to help in assigning summative judgments about a program’s merit can also be employed to improve programs. Scriven (1967) first defined formative evaluation as the use of information to develop and improve programs and products. McClintock (1984) narrowed the definition to a systematic process using “empirical procedures” for providing “ongoing information to influence decision making and action” in policy and programmatic areas (p. 77). Dehar, Casswell, and Duignan (1993) described numerous health care evaluations (e.g., Edwards, 1987) in which evaluators worked during developmental stages of programs to improve their focus, methods, and orientation. For many efforts, evaluators help programs adapt themselves to their environments by making suggestions for altering program design. Chen (1996) formally linked
process/outcomes and improvement/assessment distinctions with a two-by-two matrix of evaluation functions. “Process-improvement,” one category in the matrix, is defined as evaluation aimed at providing information for program improvement through analysis of implementation strengths and weaknesses. Chen described a process-improvement evaluation for an evaluation of a family planning program where “more married couples [were] persuaded to utilize birth control devices in an underdeveloped country if the service providers or counselors [were] local people, rather than outside health workers” (p. 124). Contextual information such as this is found only when a program meets with unanticipated obstacles in its ultimate setting. Design modifications for improving implementation (i.e., “using local people”) are generated out of contact with local conditions, an activity at the heart of formative evaluation.

**FORMATIVE/PROCESS EVALUATION OF EDUCATIONAL TECHNOLOGY APPLICATIONS**

The formative/summative distinction in evaluation roles is clearly delineated in the field of educational technology. Evaluation of educational technology has traditionally focused upon summative outcomes over descriptions of the processes leading to outcomes. Movement away from “black-box” studies, more common in other areas, has been slow to reach the area of educational technology. Bullock and Ory (2000) discussed the types of evaluation models used in higher education to evaluate educational technology, stating that “comparative studies are probably the most often used approach to evaluating technological innovation . . .” (p. 317). The huge number of studies conducted about distance education collected by Russell (2003) have in common not only their results (“no significant difference”), but also their comparative black-box design. While summative outcome studies are still used, more attention is now being paid to implementation factors that account for outcomes. In the evaluations of Apple Classrooms of Tomorrow (ACOT) (Baker, Herman, & Gearhart, 1994), and the GLOBE project (Mean et al., 1997), implementation was described and linked to outcomes. Others (Baker, 2001) have also called for inclusion of implementation evaluation for educational technologies as a method of explaining outcomes, given implementation variation among teachers using the same application, lesson, or instructional approach.

The evaluation of educational technology applications and lessons has a strong tradition of formative evaluation. Authors writing about formative evaluation in educational technology (Flagg, 1990; Gustafson & Branch, 1997; Maslowski & Visscher, 1999) agree that the practice is concerned with feedback, revision, review, and improvement of product designs. Evaluators systematically collect and communicate data (often preliminary user-feedback) to developers, and iteration and prototyping occur with different versions of an application. Additionally, designs are compared to standards (e.g., International Standards Organization, 2003) and undergo expert review. Data collection methods are generally qualitative, with extensive use of observation, interviews, and surveys, but also embrace quantitative measures related to achievement and satisfaction and measures of use, such as frequency and duration. Ideally, the evaluator works as an independent mediator between end-users and developers gathering suggestions for design improvement, synthesizing data, and communicating recommendations to designers.

While much of formative evaluation for usability takes place before an application reaches the classroom, formative evaluation of educational technologies also involves field-testing, the closest that formative evaluation comes to examining contextual obstacles. Sanders and
Cunningham (1974) were early advocates of field-testing for educational materials, calling “contextual information” a distinct category of inquiry about products that incorporates data from “the conditions under which the materials are expected to function” (p. 221). Scriven (1991) outlined four stages of formative evaluation for technology projects from early “in house critiques,” through controlled and then “hands-off” field tests, culminating in expert review.

Two types of clients typically hire formative evaluators; each has different approaches and priorities that have traditionally limited the scope of formative efforts. Software engineers (and other technical specialists) have largely concentrated on feedback about usability and interface design, with less attention given to design issues that impact the curricular and practical integration of technology. Formative evaluation of usability and interface design examines whether an educational application “works” in a purely instrumental sense, is usable, and has an attractive design (Seidel & Perez, 1994). In contrast, instructional designers and content experts evaluate lessons “delivered” by technology. Evaluation focuses on content and format to make sure that lessons effectively teach and assess their objectives (Dick, 1996). Evaluators typically make recommendations for altered wording of text, clearer directions, more appropriate lesson length and altered lesson sequence. Many projects combine elements of the two approaches as lessons are debugged, user interfaces are improved, and content and format are altered. The strategy of improving interface quality and “tweaking” of content starts early in development and continues through field-testing.

Formative evaluation of usability and lesson content/format are critical to any evaluation effort, but each set of practices may ignore how the design of applications and lessons interacts with the context of the classroom. A product with “bug-free” programming, a usable interface, and attractive graphics may still fail to be implemented because its instructional approach is irrelevant to teachers’ needs, demands a large amount of an instructor time, or requires unwieldy or unavailable technical infrastructure. Implementation concerns are generally only found during extensive and “hands-off” testing, a step that is often skipped or shortened by developers (Scriven, 1991). Even when field tests are conducted, many tests borrow from a simple product model: Like field tests of automobiles, applications are tested to learn whether they hold up under stress. Developers commonly bring with them their own artificial environments for field tests, determining where, how, and for what function an application or lesson is used. This limits the ability of teachers to fit an application or lesson into their everyday class environment, and does not allow the evaluator to report on any implementation challenges and possible fixes.

IMPLEMENTATION OF EDUCATIONAL TECHNOLOGY

What are some common implementation problems faced by teachers using applications and lessons? Formalized lessons, activities, and user-authored tools experience a complex process of integration into instruction and practice when they reach schools (Seidel & Perez, 1994). Adoption and “initial use” of an innovation involves trials with limited populations. If all goes well, the technology is then disseminated within a school district, university, or other organization unit. Ideally, an application gains a sustained foothold at multiple institutions, and is integrated into the practices of an organization. Mills and Ragan (2000) defined implementation as “the placement of an innovation in an instructional context” and described how innovations experience unforeseen use and variation in use among practitioners.

Current surveys by NCES (2000) and other studies (Hativa & Lesgold, 1996) illustrate a lackluster pattern of technology implementation in public schools. Efforts to promote
educational technology are regularly met with widespread apathy by teachers. While teachers may adopt a new application or lesson mandated for use by districts or principals, they may subsequently hide the new product in the closet after only nominal use (Cuban, 2002). Becker (1998) estimated that students averaged only 40 minutes per week of computer time, and that most computer use in the classroom primarily involves “conservative” practices, such as “basic skills” programs and games, instead of more meaningful forms of curricular integration.

Conditions for successful implementation depend partly on the beliefs, motivations, and practices of teachers. Ely (1999) called implementation a “change process” and described eight general conditions that must be in place for instructors to adopt and implement educational technology. Instructors exhibiting characteristics, such as commitment, participation, and leadership, were more likely to use educational technology. Rewards and incentives were also an important condition noted by Ely. Teachers receiving extrinsic rewards, such as extra pay, or intrinsic incentives, such as satisfaction with the accomplishment of their students, were more apt to integrate applications into the classrooms. MacArthur (1998) also found that teachers must perceive a powerful benefit from technology either in terms of learning, or in terms of improved productivity, automation, or efficiency, or both, for implementation to occur. Ertmer (1999) examined “intrinsic” implementation barriers, such as negative attitudes on the part of teachers toward technology, and found these barriers inhibit implementation even more than limited resources and access. Ertmer also found that some instructors feared the adoption of new teaching methods that incorporate the use of new technology and thus were unwilling to change entrenched teaching practices.

Organizational, physical/technical and practical factors also encourage and inhibit implementation. Easy and non-problematic access to hardware and applications is a first step to implementation, as are the timely and competent provision of technical support and effective professional development (Becker, 2000). Above and beyond these obvious external factors, Becker (1994) identified other conditions for implementation in the teaching environment of “exemplary computer users,” such as smaller class sizes and collegiality among teachers at the same school. The existence of a culture of technological use at a school also helps implementation; researchers (Bates, 2000; Rochelle, Pea, Hoadley, Gordin, & Means, 2000) found that adoption and implementation of technology became more likely when a critical mass of teachers use technology in their classrooms. Inhibiting implementation are traditional cultural practices and the organization of schools. The exigencies of classroom management and the organization of time were found to discourage teachers from changing practices to include technology (Cuban, 2002). Having enough time to master technology, author lessons, and arrange logistics is also an important inhibitor listed by Cuban.

While a wide range of implementation facilitators and obstacles are present in classrooms, many factors are not under the control of developers and evaluators of educational technology applications. Evaluators working for developers cannot change conditions found in schools, or directly change the behavior or attitudes of teachers. However, they can examine conditions that enable and inhibit successful implementation, and observe where these factors interact with product design. The remainder of the article examines technical, curricular, and practical implementation obstacles regularly found when teachers attempt to use educational technology applications and lessons. These obstacles are primarily found in the external environment of teachers, but interact with teachers’ internal perceptions of benefit, risk, and practicality of implementation. Examples of typical obstacles and recommended design changes are drawn from the evaluation practice of the author in both K-12 and higher education technology implementation.
The compatibility of an application with its ultimate setting is a major determinant of use. Designers of educational technology applications incorporate principles addressing compatibility, but the translation of a design principle to implementation tends to be incomplete because the designer cannot fully anticipate how his or her creation will function in its ultimate environment. Design that looks good on paper, in a laboratory, or in a controlled classroom environment functions differently when used outside of these contexts. Design elements also can also be a determinant of the willingness of an instructor to use new technology, and the ease in which he or she can bring the technology into a classroom.

Compatibility, in strictly technical terms, involves software fitting operating platforms, hardware, software plug-ins, and other technical infrastructure; however, technical incompatibility (in the traditional sense) is sometimes only small part of why innovations are not implemented. Other areas of compatibility, such as the organizational context of technical infrastructure, the curriculum, and instruction imbedded in an application or lesson, and the practical details of integrating new technology into a classroom may have a greater effect on whether or not an application or lesson reaches students.

Three areas of compatibility problems are presented with descriptions of common implementation problems and typical design recommendations. The role of the evaluator as an independent advisor is separate from an instructional designer and involves extensive observation of use, interviews with teachers and students, collection of logged data related to duration, frequency, and other records of use. The evaluator actively analyzes and synthesizes evidence, and then makes design recommendations. Evaluation occurs as teachers and students use applications and lessons as “naturally” as possible for instructional purposes initiated by the instructor.

**Contextual Technical Compatibility**

For the developers of new educational applications and lessons, assessing technical compatibility is usually the easiest area of formative evaluation to understand because it is similar to commonly practiced user interface and usability evaluation. Assessment of technical compatibility focuses upon the ability of applications and lessons to fit in widely varied settings, on varied machines, and with varied operating systems and browsers. Many compatibility problems are invisible if usability evaluation is only conducted in a controlled setting with ideal hardware and operating systems, but become glaringly obvious when the application or lesson is placed in a school. These problems are typically related to (1) programming and (2) mismatches between the capabilities of schools and the characteristics of an application or lesson.

**Common Implementation Problems**

**Programming.** Uncovering errors in programming is generally the job of software engineers and occurs during preliminary stages of development and analysis of infrastructure. Evaluators can help remove bugs later on in the process by reporting them to developers and encouraging them to try out their applications on as many varied machines, platforms, and operating systems as possible. Evaluators can also encourage users to report bugs directly to developers through email or website forums, and help set up and monitor this communication.
When the program is in pilot use, the evaluator can document bugs and describe the conditions present when they occurred. One difficulty with this type of formative evaluation is that programmers may insist that technology interactions with outside machines and platforms are essentially chaotic in nature. In fact, some problems are unpredictable and unrepeatable, but others will reoccur with regularity.

**Capabilities.** Obvious technical obstacles include exceptional hardware demands and problems in compatibility with existing infrastructure brought about by the design features of an application or lesson. School computer labs or home users may need specialized video cards, monitors, or processing capabilities to run applications; when faced with these demands, institutions or individuals may decide to skip implementation altogether. Highly advanced prototypical applications are most likely to pose obstacles to use in real classrooms. Teachers and schools systems may reject a new innovation if they have to make substantial purchases or upgrades in order to run the application. Many newer web-based applications run into bandwidth compatibility problems when integration is attempted in rural or economically disadvantaged school districts. At the university level, web-based modules designed for use in broadband campus labs run into the same problem of slow connectivity and limited access when used by students at home.

Evaluators can identify infrastructure variability across a collection of variables for settings where the innovation will be potentially used, and evaluate the match between the capabilities of representative schools and the features of the innovation. Variables, such as SES, school level, school setting, and school history, can be examined during early stages of design. Fortunately, some of this work is already being conducted by national survey organizations (NCES, 2000) that provide general overviews of computer-to-student ratios, types of machines in use, and other aggregate profiles of technical capabilities in public schools. However, these survey data are very general and only useful for designers making initial decisions about the features of an innovation. More customized information can be provided by the evaluator through surveys or interviews with representative technical experts in districts and focus groups with representative teachers and institutions. Most importantly, the technology needs to be frequently piloted in varied settings that represent authentic use.

**Typical Design Recommendations**

Typical design recommendations arising out of evaluation of technical compatibility include stripped-down versions of software that run on older computers and/or can be delivered to computers with lower rates of connectivity. The ability to add features optionally given the capabilities of schools is also a common design recommendation. While these are generally good design principles, they are often overlooked in preliminary development efforts and controlled field tests. Many specific modifications arising from interactions with limited capacity are also not foreseeable in the early design phase. For instance, a software designed for projecting scripted presentations of human anatomical structures worked smoothly on newer projectors used at larger universities, but needed modifications to its design to work on older computer projectors common in other, less well-funded area schools. Another application based its operation on the existence of ready access to cheap palm devices (a system that worked in controlled field tests), but proved impractical because of expense, security concerns (they were easily stolen), and their unreliability. In response, the developers altered the design to work on standard computers.
Curriculum/Instructional Compatibility

Design challenges in technical infrastructure and usability can be overcome, but what can be more potentially devastating to the adoption and implementation of innovative learning technology is poor curricular and instructional fit. Challenges in implementation and adoption were addressed by Rochelle et al. (2000), who contended that successful integration of technology occurs in combination “with new instructional approaches and new organizational structure” (p. 90). This is especially true in elementary and secondary schools where the introduction of technology is regularly accompanied by curricular reform brought about by systematic initiatives (e.g., state standards and assessments). In this environment any new infusion of technology must serve the higher goal of changing entrenched and ineffective instructional practices.

Common Implementation Problems

Content assessment. Critical to any effort is the assessment of the relevance of lesson material (if the material is curricular in the traditional sense) to the core content of standards and assessments, content narrowness or breadth, and content difficulty. Decisions about content are usually made by instructional designers or content experts early in the design process, but the ultimate assessment of fit does not occur until a lesson is implemented. Designs that incorporate imbedded instructional methods and pedagogy (e.g., “drill and practice” or “inquiry-based learning”) must also fit with current philosophies and practices emphasized by extant reform efforts. Assessment of curricular compatibility for formalized curricular products comes through direct analysis of curricular frameworks (e.g., state standards), review with curriculum experts, as well as interviews and focus groups with teachers. In cases where no instructional designer is included on the development project, information about how the innovation fits existing standards and practices can be vital to the project’s success, and it may become the evaluator’s role to provide this information through needs analysis and literature review.

Compatibility with instructional practice. Analysis of instructional compatibility in practice involves observation of the application or lesson as it is used with students, and interviews with teachers as they assess utility and compatibility with instructional goals. Applications (especially activities, simulations, and games) implicitly structure student activities, calling upon students to respond to prompts, enter answers, or conduct interactive activities. These learning activities, partly determined by the design of the application, may or may not fit with goals of the instructor, or the way in which lessons are structured in everyday classrooms. The design of the application may also resist efforts to integrate complimentary instructional activities where the application is used in an instructional sequence.

Typical Design Recommendations

Examples of design recommendations for curriculum limitations include basic alterations of content to better reflect standards, as well the incorporation of flexibility into a design so that instructors can add or modify content to fit local conditions. For example, developers of a science module for middle school students focused extensive effort on lessons in one area of content that, while usable, only fit into a limited area of typical course curricula. Instructors who previously responded positively to the content of lesson and found the lesson usable, said
ultimately that the effort involved in implementing the lesson was not worth its payoff because of its narrow focus. The recommendation in this case was to expand the content of the module to cover more topics.

Examination of how the application or lesson fits into instructional practices can lead to changes in structured lesson activities that mirror exercises and assessments found in classrooms. For instance, interviews with instructors using the anatomy software described above found that the application would work well as a stand-in for laboratory assessments using animal cadavers. While this general use was anticipated, the exact manner in which the application would be used in the classroom was not, and involved alterations to the design of the application’s interface to accommodate this use. Other design modifications related to instruction arise from the ability to modify and adapt materials. Teachers using an application allowing for automated feedback on student summaries of written text were satisfied with the functioning of the application in field tests. However, they were limited by the amount of time and work needed to implement new material in the application’s database. Design recommendations included broadening the range of preexisting texts available to teachers, and implementing a systematic and accessible method of entering texts into the database.

Practical Compatibility

Cuban has conducted a great deal of research about the practical realities of classrooms and technology and is a leading critic of the rush to wire American classrooms. In his history of technology use in the classroom (1986) and in other writings (2002), he has discussed factors in the way in which teachers’ work is organized that inhibit the use of learning technology. In Cuban’s model, teachers’ traditional practices have developed in rational ways in order to manage groups of students and use time efficiently; sometimes the use of new technology does not fit comfortably with these practices, or the limitations placed upon teachers by school structure.

Common Implementation Problems

Teachers contend with practical conditions placed upon them by the structural and organizational contexts of schools. Common practical obstacles include problematic access to technology, limited time, and the challenges of classroom management. Access is sometimes confused with the number of computers in a school, but is in reality much more complex (Becker, 2000). While a school may have a high student-to-computer ratio, the computers may be heavily scheduled, lack specific capabilities for easy use at a specific time, or be in the wrong place. Limited time, another practical constraint, is a fact of life for teachers. Any use of technology necessitates time spent in familiarization with the application or lesson, and for user-authored applications, significant time is spent creating and finalizing lessons. Teachers may also have to devote time to arranging support technology for implementation. Classroom management is concerned with how students behave because of the introduction and use of an application or lesson into an educational context. Lessons or activities must fit into existing class periods, must be easy to start, and must allow work conducted on one day to be easily continued the next. Most importantly, applications must be durable and reliable so the teacher is not left to scramble for alternative lesson plans if the technology breaks down.
Typical Design Recommendations

Given that access to hardware is different in different schools, teachers may want to use applications in a variety of settings necessitating a wide degree of flexibility and adaptability in design. For instance, a group of teachers implementing a user-authored software designed for individual student exercises could not access a computer lab often enough to make lessons a worthwhile learning activity. As an alternative they wanted to use the application for lecture, which necessitated interface design changes emphasizing visual projection. Concerns about limited time can be ameliorated through the provision of adaptable lesson plans, effective and focused professional development, and easy access to technical support. Making the interface as simple as possible can also save the instructor time and effort. For example, developers of a user-authored system designed their application to be used in tandem with an html editing program. However, few teachers had access to the html program or the time or energy to learn how to use it. The evaluator recommended simplifying the application so it could be used with a simple, more accessible word processing software. Classroom management concerns often lead to design recommendations that trade-off advanced features of an application for simpler, more reliable designs that do not break down and leave teachers without an activity for their students.

CONCLUSION

The ultimate goal of formative evaluation is to create a more usable, compatible, and effective product. To generate design recommendations, the application or lesson must be observed in its natural setting, but this step is often skipped in many formative evaluations of educational technology applications and lessons. Information and recommendations made from formative evaluation of implementation regularly focuses on non-technological phenomena, such as how teachers structure their day. Even when the focus is on the technology itself, compatibility problems usually seem almost incidental to the stated functions of an application or product, one reason why this aspect of design is so overlooked. While a product “does what it is supposed to,” aspects of the product’s design and its ability to be implemented can make functionality almost irrelevant.

Evaluators in other product development contexts, especially those where a product faces challenges in implementation, could benefit from the same general approach to formative evaluation of implementation context described in this article. The field of medical technology faces many of the same concerns as education. Lowenhaupt (2003) described failed efforts by a hospital staff to implement a new records system, and suggested those wishing to implement new systems consult with practitioners, such as nurses, to learn how the technology works in its ultimate setting. Telemedicine implementation problems were described by Stumpf, Zalunardo, and Chen (2001), who noted that most implementation problems were found in unexpected interactions between the technology and its users. Other non-educational technological applications, such as assistive technology for the disabled (NATRI, 2003) air-traffic control software (Paterno, 1998), and business communication systems (Lewis, 2003), face similar challenges with resistant practitioners, incompatibility of the product with the environment arising from preexisting practices, and modifiability in design that addresses implementation obstacles.

This article has presented a model for the formative evaluation of implementation for educational technology applications and lessons. Formative evaluation in this area has traditionally
focused on usability and curricular uses of an application, and not upon optimizing design for sustained and effective use in context. The evaluation practices presented in this article focus on generating design suggestions by learning about the technical, curricular and practical compatibility problems of an application or a lesson in its environment. Potential fixes in design are communicated to developers in an effort to make technology more compatible with the authentic conditions in schools.

REFERENCES


