Using Modular Architectures Within Distributed Learning Environments:

A Means for Improving the Efficiency of Instructional Design & Development Processes

Paul Cholmsky

A Thesis

in

The Department

of

Education

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Arts at

Concordia University

Montreal, Quebec, Canada

March 2001

0-612-59243-X

ABSTRACT

Using Modular Architectures Within Distributed Learning Environments:

A Means for Improving the Efficiency of Instructional Design & Development Processes

Paul Cholmsky

In addition to reducing costs associated with system development, modular architectures provide large-scale distributed learning environments with the flexibility required to meet the diverse needs of a global audience. Comprehensive standards for implementing such architectures are now being developed by several organizations. In order to take full advantage of these standards, instructional content design and development processes may need to be modified.

The first part of the thesis investigates the foundations of computer-based learning environments, and the historical development of instructional design methodologies in this domain. Specific emphasis is placed on the human factors involved in formalizing a design process, and how the concept of systematicity in design may evolve in the context of complex learning systems. Modular architectures are presented as a technique for ameliorating the complexity of design in this domain.

The second part of the thesis examines and critiques a recently-proposed standardized data model for distributed assessment systems. The concept of a distributed computer-based assessment system is first briefly defined. This is followed by an identification of the drivers underlying the need for a standardized data model, and a discussion of what constitutes a suitable methodology for evaluating such a standard in advance of its field implementation. After outlining the structural and functional aspects of the candidate standard, the IMS Question & Test Interoperability Information Model (Smythe & Shephed, 2000), the merits and failings of the standard are discussed, and a set of recommendations for the evolution of the standard is made.

ACKNOWLEDGEMENTS

Thanks first to my thesis supervisor, Phil Abrami, for his continual encouragement and support. I greatly appreciate the time and effort he has invested on my behalf throughout my studies in Educational Technology, and the many challenging and inspiring discussions over a wide range of topics that we have had over the years. Thanks also to the members of my thesis committee, Richard Schmid and David Wells, who have been a pleasure to work with and learn from. In addition, I would like to thank the past and present members of the Educational Technology faculty whose courses I have had the privilege of taking: Roger Azevedo, Jon Baggaley, Gary Boyd, P. David Mitchell, Steve Shaw, and Gina Walker.

A tip of the hat to Dennis Murphy of Concordia's Communication Studies department, who first encouraged me to look into graduate studies in Educational Technology, and to Anne Brown-MacDougall for her tireless efforts on behalf of myself and all other students in our department – if Dennis gets the credit for guiding me into the programme, Anne gets the credit for making sure I made it out the other side. Thanks to you both.

I would like to thank each of the Educational Technology graduate students who persevered through an entire year in the first class I taught in interactive multimedia design and development. The challenge of discussing design in an explicit manner in the classroom catalyzed some of the ideas discussed in this thesis, particularly those concerning the human factors of design methodologies.

To Robert Gordon, for the incredible three-year ride that was The Article 19 Group Inc. (version 1), and for your friendship through both the rich times and the lean. I know that you too will recognize many of the themes in this thesis, because they were first bounced around within the rubberized walls of A19. So were some of my favorite aphorisms and formulas: "No knowledge is bad knowledge"; "No one likes a two-camper"; "the FOS quotient"; "the party algorithm"; and the all-time pithy summarization of the human condition, "if it turns out that you get what you deserve in this world, does that fill you with hope or with dread?" Rob, I am confident you will get what you deserve.

To my fiancée, Laurel Eakin, who puts up with me on a day-in / day-out basis. This fact in and of itself is likely of great amazement to most people. Couple it with endless intellectual stimulation, a bottomless well of emotional support, and oodles and oodles of laughter, and I feel like I am the luckiest man (or gorilla) on Earth. Now if we could just find muzzles for the cat and the singing asparagus, all would be well...

Finally, to my mother and father, who from Day 1 have always made sure that I have had the opportunity to pursue my studies. Without their many sacrifices over the years, this thesis – and many other things that I have been the lucky benefactor of – would not have come to be. With each passing year, I realize more and more the profound importance of family in one's life. Tragically, my father passed away during my graduate studies, and is not here to see the final result. I dedicate this thesis to his memory, and hope that it would have made him proud.

This research was funded by Fonds pour la Formation de Chercheurs et l'Aide à la Recherche (F.C.A.R.), and a fellowship from Concordia University.

TABLE OF CONTENTS

The Evolution of Computer-Based Learning Environments in Educational Technology

Educational Technology on Internet Time

Educational technology is a domain of research and practice that is characterized by plurality. Saettler (1990) believes the field is in a "preparadigmatic" (p.15) state, in that no single Kuhnian paradigm can be identified that characterizes it. This diversity exists at numerous levels: pedagogical strategies can draw upon behaviorist, cognitivist, and constructivistic theoretical models (Ertmer & Newby, 1993); instructional design models can be categorized on the basis of whether they are classroom, product, or system oriented (Gustafson & Branch, 1997a); and quantitative, qualitative and systemic research methodologies each have their proponents (Salomon, 1991; cf. Shaw, 1990). In fact, it could be argued that no single tenet exists that is shared by all members of the educational technology community, or even a majority.

From one perspective, this plurality is useful. As a domain that, at the end of the day, must be concerned with accomplishing interventions that effect that fuzzily-defined state known as learning, educational technology requires a diverse and flexible toolbox to accommodate the variety of contexts it is called to bear within. Indeed, the Greek root of technology, <u>techne</u>, encompasses both the systematic application of objective knowledge as well as more subjective individualistic approaches characteristic of the craftsperson and the artist (Saettler, 1990). In order to not force a harsh divide between the artistic and the scientific cultural mindsets (Snow, 1969), a design domain may need to limit the extent to which a dogmatic 'technical' rationality is given a sort of a priori supremacy (Schön, 1983). Furthermore, a plurality of perspectives ensures a healthy debate, and can be taken as a marker that educational technology's scope of enquiry is rich in its breadth and depth.

The potential downside to this state of affairs is that research can become increasingly divorced from practice, as the diversity of issues investigated by the former becomes increasingly irrelevant to the immediate concerns of the latter. In his seminal article entitled 'You can't play 20 questions with nature and win', Newell (1973) pointed out that scientific progress itself can be hampered when the search for answers to cogent questions takes a backseat to the continuous iterative splintering of relatively obscure theoretical issues. As an applied science, educational technology has an even greater responsibility to field practitioners to stay relevant in a tangible, concrete manner.

In the subdomain of educational technology that focuses on the development and implementation of computer-based products, the situation may be substantially more problematic, because the base materials in this design domain have evolved so rapidly as of late. Moore's Law has created a situation wherein the cycle time of conventional research enquiry is grossly mismatched with the ever-shortening cycle time of technological innovation. Researchers and developers, like everyone else, face the challenge of updating their practices in order to compete on 'Internet time' (Cusumano & Yoffie, 1998). Without this evolution of practice, researchers may face the obsolescence of many of their technology-dependent conclusions before publication, short-circuiting educational technologists' ability to build upon and progressively refine their field's knowledge base. Instead, what is left is termed by Tennyson (1994) as the 'big wrench' approach, where single researchers or small groups working in isolation individually search for a singular, self-contained theory that can have a large, generalizable effect on instruction. The lack of a shared framework guiding these nodes in the research network limits potential innovation to the resource limitations of each individual node.

In addition to an exponentially increasing rate of change, a further complicating factor in this design domain is that its software and hardware components are becoming increasingly complex. One approach for managing this complexity is to search for the common aspects of computer-based learning systems and identify their conceptual underpinnings. By focusing on those that lie at the slower-moving core of this design domain as opposed to its faster-moving periphery, the conclusions obtained may apply to several technological generations rather than solely the most current. In pursuit of this goal, the next section of the thesis explores the historical antecedents and fundamental concepts of complex computer-based learning environments.

<u>Revisiting the Teaching Machine</u>

Buck (1989) defines the teaching machine as "a mechanical, pneumatic, electrical, electro-mechanical, or electronic device which, upon some sort of manipulation (input) by the user, performs some sort of transformation of the input and then provides some recognizable form of instructive feedback" (p.32). The teaching machine is thereby distinguished from a teaching aid, such as a writing tablet, which does not generate feedback based on transformation of learner input. Thus, Archimedes' mechanical simulations of celestial motion for instruction in astronomy (circa 250 B.C.) are teaching machines, since learners could manipulate components of the model to explore emergent phenomena such as eclipses, whereas Thales' earlier three-dimensional but static representation of constellations (circa 550 B.C.) is a teaching aid. The concept of a teaching machine qua mechanism for generating instructional feedback is more precisely exemplified by Buck through a distinction between the palus and the quintain, two devices used by the Romans in military training. Essentially a vertical pole affixed in the ground, the palus merely provided a stationary target for weapons practice. Roman military instructors therefore still had to closely monitor trainees' interaction with this device to prevent poor technique from being practiced and ingrained. The more elaborate quintain added a punitive device of some sort (such as a blunt weapon) which was attached to a target on a revolving horizontal rod so that incorrectly delivered strikes on the target would cause the punitive device to revolve towards the trainee at speed. By automating the delivery of feedback, albeit in a relatively harsh manner, the quintain absolved instructors of the obligation to constantly supervise trainees on an individual basis. It is arguably then the first example in educational technology of an artifact that was borne out of a need to improve instructional efficiency as well as effectiveness.

The wartime roots of modern education technology are of course not limited to the early innovations of the Romans. After the Second World War, Bush (1945) published an influential article entitled 'As we may think' in The Atlantic Monthly wherein he enjoined scientists to derive peacetime applications from the numerous technological advances achieved through military research. A substantial part of the article is devoted to a description of his vision for a device called the 'memex', which would allow both authors and readers to establish links between discrete units of content. These links, persistent over time, could subsequently be used by anyone to quickly retrieve the associated content at any time in the future. Bush believed that the ability to more adeptly gather, organize and retrieve information on a large scale would have a transformational effect on society's ability to generate, exchange and acquire knowledge. He also felt that the ever-increasing amount of scientific information made such a device not only desirable, but necessary. The proposal for the memex, and its concept of associative indexing, contained the seed of an idea which would decades later manifest in such ubiquitous data storage and communication technologies as relational databases, hypertext, and the World Wide Web.

From the perspective of educational technology, there is a core facet of the memex's design that is of especial interest - the concept of metacontent, or 'content about content'. In a primal sense, the first pedagogical application of metacontent can be traced to the 17th century, with Comenius' suggestion that the top-level outline for a given course of study be continually displayed on the walls of the classroom or lecture hall (Saettler, 1990). This precursor of more sophisticated instructional aids such as the advance organizer (Ausubel, 1968) is also reflected in Bush's vision of the memex as a learning tool. One of the hypothetical examples he provides of the memex's operation describes an individual studying the history and material properties of the bow and arrow. In outlining how the student could use the memex to navigate and manage a large set of disparate learning resources, Bush primarily argues for the advantages specific to the explicit separation of discrete units of content from the physical structure of the artifact that the set of content units is embodied within. Authors and readers would no longer be constrained to a single pre-specified linear path through a given body of content, a path that was made manifest by and bound to the physical arrangement of leaves in a book. Instead, the memex would enable the equal pursuit of a multitude of trails whose branches were continuously defined, evolved and shared by a community of author-readers. Therefore, the core of Bush's explicit argument is that the memex would increase both the practical scope of materials that could be utilized within a given learning session, as well as the total amount of materials available given that the distinction between authors and readers would be partially erased. By enabling the reader to add descriptive commentary about existing content units to facilitate their future retrieval, however, the memex also demonstrates a mechanism for adding metacontent to the system. The memex is therefore perhaps the first example of an educational technology that implicitly separates metacontent from content, in terms of enabling learners to use the metacontent to differentiate between content units. By describing this mechanism, Bush is also making a case for the value of metacontent in a collection of low-grained content units.

A little over a decade later, Skinner (1958/1996) argued for the pedagogical advantages of the 'teaching machine' by name. Originally proposed and developed by Pressey in the 1920s and subsequently further realized by Skinner, these teaching machines were devices which contained a large number of distinct units of instructional content called 'frames'. Each frame contained a question or problem for the learner to respond to. Learners would view these frames one at a time, and the teaching machine's selection of the subsequent frame was based on the correctness of the learner's response to the question posed on the preceding frame. The teaching machine's sole role was to allow the learner to view the frame, process their response, and then execute the pre-specified branching strategy. Skinner's enumeration of the factors contributing to the teaching machine's pedagogical value included the following:

- Learners play an active role in the instructional process, rather than being passive receivers of information as is the case while attending classroom lectures, reading textbooks or watching films
- Immediate, individualized feedback is provided to the learner on the quality of their responses to questions posed
- Mastery learning of instructional content is promoted by supplying learners with numerous opportunities to practice their newly-acquired knowledge/skills and requiring them to demonstrate a prespecified level of competency before progressing onto more advanced topics
- Learners can progress at their own pace through instructional materials
- Instructional content is selected and sequenced based on each learner's ability and the outcomes of previous interactions between the learner and the artifact
- Learners' access to instruction can be increased without requiring the availability of additional tutors, classrooms, or other students
- The best practices of human tutors are emulated, such as providing support and guidance to learners (e.g., hints and prompts) during the learning process
- Learners are more engaged by the instructional material and their motivational levels are higher

The similarity of this list of benefits to those frequently cited for more modern computer-based technology is striking. It is important to bear in mind that the hardware employed in the artifacts designed by Pressey, Skinner and Bush is decidedly archaic by contemporary standards. These teaching machines were not crafted from integrated circuits and silicon chips, but rather from gears, levers, and photoelectric cells. In this regard, they share a greater kinship with clocks than with today's digital computer. These clockworks, however, provide the initial forms of key concepts at the heart of the computer-based learning system: the automated transformation of input and delivery of feedback in service of both greater instructional efficiency as well as effectiveness; the explicit separation of program from hardware, and message from container; and the dual nature of instructional content 'units' in these systems as dynamic and static entities. Thus, in considering the modern distributed learning environment, we are seeing the synthesis of a number of core concepts that in a certain respect have evolved over centuries rather than decades.

Perhaps the evolution of teaching machine hardware is only of academic interest, however, without real relevance to practitioners. Indeed, some academic researchers have even argued that the more 'technical' aspects of hardware in general lie outside the province of educational technology. In the next section, the merits of this argument are considered.

The Medium, the Method and the Message

In considering the historical importance of Skinner's teaching machines, Saettler (1990) argues that the overt emphasis placed on the teaching machine as a hardware artifact was misplaced. He believes that its true significance lay in the development and evolution of programmed instruction rather than the design of the teaching machine itself per se. As a warrant for this statement, he cites the relative proliferation of programmed textbooks in order to make the claim that the pedagogical strategy employed was not exclusively bound to machine-based mediation. This line of argumentation is similar to that employed by Clark (1994) to reach the more general conclusion that media in and of themselves cannot influence learning. In Clark's view, educational media are "mere vehicles" (1983, p. 445) for instructional content in the same sense that grocery trucks are merely vehicles for the distribution of groceries. To wit, with respect to the goal of improving people's nutritional levels, it is the food that they eat rather than the manner in which it is distributed that makes the difference. Kozma (1994) partially rebuked this argument by pointing out that different media will have different levels of suitability for a given pedagogical strategy, and thus reframed the media / method argument in terms of the relative fit between a given medium and a given method. The debate, however, continues on, most noticeably in Russell's (1999) much-discussed conclusion of 'no significant difference' between traditional face-to-face classroom instruction and distance education based on the results of over 400 studies.

As Ehrmann (1995) comments, both of the perspectives put forward by Clark and Kozma have their respective merits, and each is useful in discussing media and method both in isolation and in association. Per Clark, one can conclude with reasonable safety that any given medium is likely neither a necessary nor a sufficient condition for accomplishing a given type of instructional objective. Per Kozma, one can conclude that for a given instructional strategy, there may likely exist certain media which are demonstrably more appropriate for implementation of that strategy than other media. The danger lies in stopping the analysis here, thereby belittling the potential role of media research in educational technology by restricting it to "narrow efficiency questions" (Clark, 1984, 281). As McLuhan (1964) has observed, the advent of both rail and air transportation had pronounced social effects independent of the specific cargo they carried. In the same vein, the use of computers in educational settings may have more general effects independent of the subject matter of the software that runs on them. In a less abstract sense, an argument could be made that although grocery trucks are not directly implicated in nutrition, the use of grocery trucks for food distribution changes the kind of food available at the local grocer's. Analogously, part of the impact of teaching machines on learning may be through the economics associated with content development and use.

In the face of the ghettoization mandated by Clark, however, the study of these economic factors can seem to be of trivial importance. The ever-increasing level of sophistication in the design of computer-based instructional systems has resulted in noteworthy gains in terms of their pedagogical effectiveness (e.g., Lajoie & Lesgold, 1989; Schofield, Evans-Rhodes & Huber, as cited in Derry & Lajoie, 1993; Anderson, Corbett, Koedinger & Pelletier, 1995). From one vantage point, it could therefore be argued that steady progress is being made towards the development of computer-based tools for learning that will match the high level of efficacy associated with one-on-one tutoring by human tutors – the widely-cited two-sigma problem as posed by Bloom (1984). In brief, the two-sigma problem is the search for "ways of accomplishing (the two standard deviation gain in achievement attributed to human tutoring) under more practical and realistic conditions than one-to-one tutoring, which is too costly for most societies to bear on a large scale" (Bloom, 1984, 4). Successfully attaining - and potentially surpassing - this lofty goal, which has been referred to as the 'gold standard' in research on intelligent tutoring systems (McArthur, 1992), may seem like only a matter of time (Advanced Distributed Learning Initiative, 2000).

From a different perspective, however, the state-of-the-art in computer-based learning system design can equally be depicted as relatively unsophisticated (Self, as cited in Reye, 1996). Undesirable consequences of the increased complexity of technologically-advanced learning systems have included a substantial increase in their development cost, decreases in the flexibility and adaptability of these systems, and decreased maintainability (Ritter & Koedinger, 1996). As a result, the impact of computer-based learning technology on large-scale educational and training systems as a whole has been limited to date (Advanced Distributed Learning Initiative, 2000). For example, although many educational institutions are offering WWW-based courses or complement classroom-based instruction with WWW-based interactive resources, these efforts rarely scale to the institutional level in terms of creating a coherent institutional program for implementing technology-based education (Graves, 1999).

In deciphering this puzzling state of affairs, it is important to bear in mind the crux of Bloom's (1984) 2-sigma problem; namely, that society already possesses an effective instructional technology (i.e., one-on-one tutoring by human tutors), but that there is a prohibitively high cost associated with employing this technology within an educational system. The two-sigma challenge therefore speaks directly to both the efficacy <u>and the efficiency</u> of a given instructional resource. R&D initiatives in automated learning systems, while making significant progress towards replicating the effectiveness of human tutoring, have also replicated its high cost, and thus can at best be considered to have only partially met Bloom's challenge. Indeed, the high cost of complex learning systems has been identified as a principal barrier to their widespread deployment (Blumenthal, Meiskey, Dooley, & Sparks, 1996). In order to fully address Bloom's challenge, there is a pressing need for designers of complex learning systems to focus on the efficiency and scalability of their development processes as well as the effectiveness of the outputs that are produced by them. As noted by Bharwaj, Chandran, O'Neal & Gibbons (1998, p.102), "the day of scaling-up for computer-based instruction is upon us".

Perhaps educational technology shall smoothly transition into a phase where the economics of instructional content development are given due consideration. For example, scalability and other 'technical' aspects of technology deployment are increasingly being considered as necessary components in grant proposals (e.g., National Science Foundation, 2000). In the literature, Kozma (2000) identifies issues related to the scaling up of instructional design projects as one of the five new themes that define the forefront of research and development in educational technology. In a survey of consulting editors for the leading journal Educational Technology Research & Development, among the top five preferences for topics were studies of instructional design, development, and evaluation processes, instructional design models for computer-based distributed learning environments, and the general innovation and adaptation of instructional design models (Klein, 1997). In this vein, Richey (1998) calls for focused research on the reduction of design cycle times, resource utilization and management issues, development techniques such as rapid prototyping, and designers' decision-making processes.

A potential barrier to this smooth transition, however, may lie in the traditional conception of instructional content development itself. Tripp & Bichelmeyer (1990) have noted that systematic instructional design as a whole emphasizes effectiveness over efficiency. Researchers and research-based prototype developers have a tendency to ignore costs in their search for perfect educational paradigms, resulting in products that are too expensive to produce, use and maintain (Lesgold, 1998). As there has also been relatively little validation of instructional design models' efficiency or effectiveness (Gustafson & Branch, 1997b; Richey, 1997), there are few established practices for doing so. To rectify this shortcoming, Richey (1997) has advocated increased research into content development processes, tools and models through case studies, quantitative experiments, and historical and philosophical investigations. In Richey's view, the quality criteria for this research program must focus on its utility for field practitioners; descriptive and explanatory inferences must lead to the prescription of replicable procedures and techniques. As noted previously, however, an emphasis on the economics of content development may face resistance from researchers who believe a priori that 'media cannot have an effect on learning'.

Another potential barrier to a smooth transition is the degree of fit between current instructional design and development models and the design domain of complex learning systems. Large-scale distributed learning environments, responsible for configuring, adapting and delivering dynamic content across a multitude of organizations and platforms, may involve design issues so different from classroom instruction, print-based documentation, and even traditional computer-based training that the gradual evolution of existing approaches will prove insufficient. In the conclusion to their comprehensive survey of instructional design models, Gustafson & Branch state that, "clearly, current ID models would be too limited for thinking about, developing and maintaining ... environments" (1997b, p. 86) involving dynamic, database-driven content. It is an open question whether processes primarily developed in the context of authoring content in small-scale closed systems will evolve smoothly into processes for architecting large-scale distributed learning environments where the number of distinct content components may be in the hundreds or thousands (Cholmsky, 2000).

It could be argued that a substantial portion of the new demands placed on instructional design and development models in this context is of a primarily technical nature. The challenge posed by the confluence of pedagogical and technical design issues may therefore be partially solved by interdisciplinary design teams, but the line dividing these issues is fuzzy in many cases. Consider the development of a set of interactive practice exercises that include variable remedial feedback based on learner response patterns (Cholmsky, 2000). A design decision could be made to use a database-driven approach wherein feedback is assembled from low-level content components that are sourced from a database. If these low-level components include subject-matter terms and their definitions, the same database might be used to drive a glossary function in the learning environment, serving the objective of content reusability. In designing such a system, we would likely not expect an information technology professional to author the subject-matter definitions that lie within the database, just as we would not expect the instructional design professional to design the software that connects the ODBC database to the content delivery system within a client-server environment. Appropriate allocation of responsibility for many other design issues, however, is not as clear-cut. For example, is one of these professionals primarily responsible for decomposing the feedback into suitable lower-level components, or is this a joint responsibility - and, if so, how do these professionals collaborate on this task effectively? Of specific concern are the pedagogical implications of successful technical convergence (e.g., the use of database-driven content systems, or the adoption of XML as a markup language): does this form of standardization support research on and development of pedagogical strategy, or does it sacrifice flexibility at this level to achieve the goal of technical convergence? Discussion of these issues requires a clear articulation of the interface between pedagogy and technology in practical terms.

It would be comforting to disassociate educational technology from such seemingly arcane issues as relational database design, interoperability standards development, and three-tier system architecture. After all, what do they have to do with students' learning? For those who share Clark's view regarding media and method, perhaps nothing. The end result of this disassociation, however, may well be another generation of educational technology that has relatively limited impact on the real-world conduct of education and training. Granted, educational technology is not computer science, but then again it is also not cognitive science, educational psychology, or organizational management. Educational technology professionals do need, however, to draw on these domains and many others in their practice. Mastery of them all is of course an impossible requirement. For educational technology to serve society in a practical sense, it needs to refine its interfaces with its associated domains, and provide usable conceptual black boxes that convey sufficient understanding for the working practitioner to utilize the tools provided by these domains for the purposes of promoting learning.

<u>Summary</u>

In this introduction, the following arguments have been made:

- Educational technology is characterized by a plurality of orientations, methods and assumptions. An unfortunate consequence of this plurality is that research is to a certain extent disassociated from practice.
- Although the increasing pace of technological innovation has provided researchers with a number of potential 'big wrenches' for improving learners' performance, significant impacts on large-scale educational and training systems remain limited. To remedy the conceptual myopia that Internet time can incite, it is important to bear in mind that the core principles embodied in the modern distributed learning system predate the arrival of the integrated circuit.

- The efficiency of a learning system, both in terms of its development and in terms of its deployment and operation, must be considered equally with its effectiveness. Otherwise, the high cost, limited flexibility, and design complexity of technologically-advanced computer-based systems will significantly limit their practical impact.
- The current practice of instructional design, predicated on assumptions relevant to traditional delivery media such as classroom instruction, print-based documentation, and standalone computer-based tutorials, may be insufficient to meet the demands of designing complex computer-based learning environments.
- Research on new instructional design methods must be undertaken to meet these demands, and this research should result in models, techniques and procedures that can be directly implemented in the field.
- Although a simplistic conception of a causal relationship between the use of a given medium and learning is rejected, the technical aspects of computer-based media must be considered in order to better understand their relationship with pedagogical aspects.

The next section of this thesis investigates how design processes in educational technology can be evolved through the use of modular architectures.

Improving the Efficiency of Learning System Development via Modular Architectures

Complex learning systems are commonly developed in relative isolation from one another. For example, there is no sharing of base-level components between systems in the community of researchers and developers working with adaptive learning systems (Winne, 1993). A possible strategy for reducing the time and cost of developing complex learning systems, however, is to create a context wherein whole systems would not have to be designed largely from scratch (Reye, 1996). A key aspect of such a strategy would be the establishment of cross-organizational standards that would allow developers to reuse content components from other systems, both from within their own organization and from external sources (Rowley, 1996). Recently, coordinated proposals have rapidly emerged from several bodies as candidate standards for this purpose (e.g., Anderson, T., McArthur, D., Griffen, S. & Wason, T., 1999; Advanced Distributed Learning Initiative, 2000). These standards represent an attempt to replicate the productivity gains from reusability and interoperability that have been achieved in software engineering (Sodhi & Sodhi, 1999) within the domain of instructional content development. This follows a general trend of looking towards software engineering for inspiration in improving the efficiency of instructional design processes (e.g., Maher & Ingram, 1989; Tripp & Bichelmeyer, 1990; Wilson, Jonassen, & Cole, 1993; Lesgold, 1998). Preliminary estimates of the potential cost savings from reusability and interoperability in learning system development range from 50-80% (Advanced Distributed Learning Initiative, 2000).

In a later section of this thesis, one of these candidate standards is discussed in depth. Prior to entering into this discussion, however, it is necessary to investigate the fundamental concepts that determine the possibility of reusability and interoperability in instructional design projects. The first aspect in this regard is the concept of design decisions and dependencies between design decisions.

Codependency in Design Decisions

Consider an instructional design project where a textbook is to be written, and the desired timeframe for the completion of the project is shorter than that estimated for completion by a single author. One reasonable approach would be to divide the textbook into two sets of chapters, find a second author, and have each author concurrently write one of the chapter subsets. Although the overall development time would still be equal to the longer of the development times of the individual chapter subsets, the operative assumption would be that this would be a shorter time than either author could finish the entire textbook in.

Under what circumstances would this be a reasonable assumption? An important consideration here would be whether each author could independently make decisions about aspects of the chapters he or she was writing without affecting the other author's work. For example, if the textbook consisted of a collection of chapters on relatively diverse topics, then it would likely be that each author could have relative freedom to develop the treatment of each chapter's topic. This would probably be the case if it were true that chapters could be added, deleted or their topics substituted without noticeable adverse consequences on other chapters.

A contrasting example would be a textbook wherein each chapter builds on content that came before it, with basic topics covered in the early chapters of the book and advanced topics treated in later chapters. In this case, dependencies between the authors' work will exist wherever one author has a responsibility to cover a basic topic before the other author can cover a more advanced topic that builds upon it.

In considering dependencies of this nature, it is important to first clarify how they are distinguished from the vast set of identifiable similarities between artifacts that are designed to function as components of a larger system (such as chapters in a textbook), and then to establish a means for identifying them within the set of design decisions made on a project. It may be useful to distinguish dependencies from perceivable similarities by associating the former with decisions made during the design of an instructional artifact. In the case of a textbook on some form of mathematical analytical procedure, for example, it may be true that a sole standard notational system is used throughout the field by professionals and educators for representing the procedure. If this is so, then it could be expected that all textbook authors would employ this notational system in their textbooks. In a project where multiple authors were collaborating on a single textbook partitioned into chapters, it would likely follow that the single notational system would be used in their respective chapters. Although the notational system could be considered as an identifiable similarity between the authors' sets of chapters, it is not a dependency in terms of design decisions, since no decision was made here.

The case would be different, however, if a single standard notational system was not in existence, or the authors of a given textbook planned to depart from convention and develop a new system or systems, perhaps on pedagogical grounds. In this context, the design process would require one or more decisions by the authors about the notation to be used in the textbook, and the possibility would exist that two or more authors would make different decisions, if these decisions were made independently. In this scenario, dependencies may exist, because design alternatives exist.

The existence of alternatives in design, however, does not automatically imply the existences of dependencies between designers' decisions. Returning to the example of a textbook on a mathematical analytical procedure, there are two possible outcomes if a notational system is part of what is 'designed' with respect to the textbook. The first possibility is that, whether by mutual agreement between the authors or by chance, the same notational system may end up being used throughout the book. The second possibility is that two or more different notational systems may be used. In each of these situations, there is an additional distinction that needs to be made. Assuming that multiple notation systems end up in the textbook, it may be that:

- the use of different notational systems in different chapters does not affect any quality criteria related to the book as a whole. In this case, each author is free to optimize the notational system used in a given chapter to the needs of that particular chapter (i.e., to the local quality criteria of that chapter). This can be stated generically to be the case when the definitions of a design parameter and its associated quality criteria do not cross the boundaries that demarcate each author's separate scope of work.
- the use of different notational systems per chapter has an impact on one or more quality criteria related to multiple chapters or the book as a whole. This might be the case, for example, if the use of several different notational systems negatively affected the textbook's overall effectiveness. Perhaps the need to learn and apply more than one notational system causes frustration and confusion to be experienced by readers. Note that both the design parameter (i.e., the use of more than one notational system in the textbook) and the quality criterion (i.e., the textbook's overall effectiveness) are articulated across the boundary that divides the designers' individual works.

These non-local design parameters and quality criteria automatically create dependencies between the design decisions that impact them.

This distinction between local and non-local design parameters and quality criteria in terms of dependencies also extends to the former case where the same notational system is employed throughout the textbook. It may be the case that using the same notational system in each chapter detracts from the overall effectiveness of the textbook. Different notational systems may emphasize different aspects of a given mathematical procedure, and the use of a single representational scheme may therefore negatively impact a reader's ability to develop a multifaceted understanding of the procedure.

An interesting example of this type of phenomenon can occur in instruction on the nature of current flow in electrical circuits. Gentner & Gentner (1983) found that students who used a model of 'moving crowds' as an analogy for current flow exhibited systematic differences in reasoning about circuits compared with students who used a mental model of 'flowing fluids'. Each of these analogical models provided superior support for a different pattern of inference. For a more complete understanding of electrical circuits, it may therefore be advisable to use both models in instruction in order to develop learners' analytical flexibility (cf. Jacobson & Spiro, 1995), enabling them to employ the one that is most appropriate for analyzing a given circuit segment. What this hypothetical example demonstrates is that consistency across design parameters cannot be universally equated with optimizing quality criteria and thereby achieving maximal design quality. In contrasting cases where consistency in interface design is beneficial with those where it is detrimental, Berkun (1999) reminds us of Emerson's poetic summarization of the same idea: "Foolish consistency is the hobgoblin of little minds".

A definition of design dependencies must consequently avoid specifying a value for the design parameters implicated in it. A design dependency is therefore considered here as any kind of relationship between design parameters and quality criteria that cross some boundary within the design, regardless of what values the parameters must hold in order to optimize the design on that criterion. In the case of the notational system(s) employed in the textbook, the decisions of each designer in this regard are dependent on one another because the definition of the implicated quality criterion crosses the boundary of their respective scopes of work.

If the majority of significant dependencies in a design domain are well known or can be identified a priori, a front-end planning phase may be sufficient to satisfy the coordination of subsequent design tasks in the project. In a textbook development project, this would be the case if the design parameters (e.g., which topics to include, what chapter structure to use, which notational system(s) to employ, etc.), quality criteria (e.g., subject matter or domain coverage, instructional effectiveness, overall page count, etc.), and their interrelationships were well understood at the outset of the project. Under these conditions, it should be feasible to conduct an analysis during a front-end phase to derive usable lower-level design objectives and constraints for each author's chapter subset from the top-level objectives and constraints for the project. During this front-end phase, the dependencies between chapter sets would be identified, and values would be set for their implicated design parameters. This information would be communicated to each author, allowing them to develop their content sections with a comprehensive understanding of their section's functional and structural responsibility to the overall work. Without this form of front-end analysis and project management at the system level, the potential for incurring extensive rework downstream exists, since authors working in parallel may easily create chapter subsets with conflicting values on numerous design parameters. The project can then end up in the unenviable position of taking longer than it would have if only one author was originally assigned to the entire development task.

Front-end planning, even when conducted in an ideal manner, is not a universally applicable solution. In the design of a complex artifact, large networks of interactions between design parameters and quality criteria can overwhelm a designer's or team's ability to plan and accommodate for them in advance. Another aspect of this problem is that, in addition to the initial analysis of a project on the basis of a conceptual model of the system to be designed, designers and design teams must also deal with the evolution of design knowledge about the system during the course of the project. In many cases involving complex systems, it is primarily the substantial amount of new information that is uncovered during a typical development project that leads to high levels of complexity in design:

- new quality criteria are discovered, existing ones evolve or are discarded, and these modifications in quality criteria lead to new networks of dependencies
- new design parameters are discovered and made explicit, potentially leading to new dependencies

- new relationships between known design parameters and quality criteria are discovered

To illustrate these categories, the design of a hypothetical educational WWW site will be considered, wherein novice designers are unaware of usability as an explicit quality criterion at the outset of a project. Given the lack of explicit consideration of usability, it will assumedly not factor into the initial stages of the site's design and development. Midway through the project, however, preliminary field tests may indicate that learners are expressing low levels of what Nielsen (1993) terms 'subjective satisfaction' with the site's navigational scheme, and that a significant amount of them are sufficiently frustrated that they cease to continue using the site. This is an example of a new quality criterion emerging during the course of a design. The designers must now reassess the design and identify those design parameters (e.g., type of controls used for navigating through the site) that are negatively impacting usability and therefore subjective satisfaction in this regard.

New design parameters may also emerge during the course of a project. The designers of the hypothetical educational WWW site may have initially planned to use audio narration to accompany animations of causal processes. Thus, they could be said to recognize relatively broad decisions about media as a variable parameter in the design of a WWW site. In actually developing the site, they may realize that an additional decision must be made regarding whether the audio narration will be presented concurrently with the animation or if the presentation will be sequential (e.g., audio narration first, and then the animation). This is an example of an emergent design parameter. If the designers wanted to follow a consistent format between different sections of the WWW site, then a decision would have to be made about the contiguity (Mayer, 1997) of these media elements, and this decision would need to be communicated to the developers of each section of the site.

Even if design parameters and their associated quality criteria are known and explicit at the outset of a project, new interactions may still emerge during its course. For example, the designers of the hypothetical educational WWW site may know from previous experience that a decision about the contiguity of audio narration and animations must be made when planning a WWW site. They may also be planning to employ a metric measuring learners' gains in problem-solving ability after using the WWW site as part of their quality criteria to evaluate the site's design. If they are unaware, however, at the beginning of the project of a relationship between contiguity and subsequent problem-solving ability (cf. Mayer, 1997), then the initial decision about contiguity will be made on other grounds. Perhaps the rationale will be technical: it may be problematic to coordinate the synchronized playback of the animation and the narration, if they exist as separate media files. If the animation's appearance is seemingly enhanced by the existence of concurrent audio elements (Laurel, 1992), then the decision about contiguity may be made initially on aesthetic grounds. Later in the project, as a result of pilot testing and subsequent experimentation with the design, a relationship between contiguity and learners' problem-solving ability may be identified. As a result, the design will have to be reevaluated to determine whether any previous design decisions about contiguity need to be revisited. Furthermore, there may be a need to balance the pedagogical, aesthetic, and technical perspectives on the design if there are conflicts between them, and explore possible tradeoffs. In this sense, information about relationships between known design parameters and quality criteria may emerge as a project progresses.

These examples serve to demonstrate that the effective management of complex design projects, especially those involving multiple designers, requires a mechanism for identifying design parameters, quality criteria, and their interrelationships, and coordinating project work on this basis. This mechanism needs to function during both the initial planning of the design effort and as it runs its course, in order to handle both a priori and emergent design knowledge. This is an important point, because an argument is not being made here that any kind of front-end analysis is de facto precluded by the complexity of instructional design within technically complex learning environments. This oversimplification can lead, for example, to the equivocation of a rapid prototyping approach to development (Tripp & Bichelmeyer, 1990) with a complete absence of front-end analysis. That is, there is an unfortunate tendency amongst inexperienced educational software developers to defend the lack of even the most preliminary design rationale through the false justification embedded in the phrase "but we're rapid prototyping". This tendency is symptomatic of an excessively coarse distinction between primarily iterative approaches such as rapid prototyping and primarily linear, sequential approaches such as phased waterfall models, relegating designers to an either/or decision in the face of this illusory dichotomy. In determining the appropriate design management approach for a given project, several factors will need to be balanced: the extent to which the design domain is well defined; the number of designers working on the project; the cost of rework, and so forth.

This balancing act is not endemic to the design of complex computer-based learning systems alone. In discussing the management of testing processes in educational film development, Rose & Van Horn (1956) stressed the need for a conscious tradeoff between the increased economy that results from moving revision upstream and the increased accuracy that results from moving revision downstream. Another important concern, found in any design domain, is the relative prioritization of development cost, delivery time, and product performance (Reinertsen, 1997). While exploring the concept of a 'layers of necessity' model for instructional design, Tessmer & Wedman (1990) touched on exactly this point, in that designers need to tailor their overall methodology to their local context. As they put it, designers can optimize their development processes on any two of 'better, faster, or cheaper', but not all three. Once these broad objectives have been investigated, project managers are in a better position to select the correct framework for the design endeavour at hand.

In the next section, the role of system architecture in providing such a mechanism for design project management will be discussed.

Modular Architectures

Fundamentally, modular architectures can be viewed as conceptual structures for addressing the needs of designers in the analysis and management of complex design projects. In the computer hardware industry, for example, the development of IBM's System/360 family of computers in 1964 is regarded as the first instance of a modular architecture in this design domain (Baldwin & Clark, 1997). Prior to that year, each component of a given IBM computer model was designed specifically for that model from the ground up. As a result, evolutions of the product line frequently required the redesign of most of the product components, both in terms of hardware (e.g., processor, peripherals) and software (e.g., operating system, application software). In the case of the System/360, however, the product designers departed from this design approach by envisioning a family of computers that could be tailored to different markets while sharing a common set of hardware and software components. One of the primary benefits of this approach was that a small group of senior designers could effectively translate the top-level design goals for an entire product line into a single modular architecture, which was used to efficiently manage dozens of globally-dispersed design teams responsible for producing specific modules in the architecture (Baldwin & Clark, 1997). The outputs of these design teams could then be seamlessly combined into complete functional products. This conceptualization of the final product as a combination of distinct 'building blocks' that are designed independently but act together to achieve an overall function is a common characteristic of a modular design approach (Stone, Wood & Crawford, 2000). The System/360's modular architecture is considered to have been a key factor in the commercial success these computers enjoyed, and was to have a significant impact on the computer industry as a whole (Baldwin & Clark, 1997).

A modular approach to design can therefore be contrasted with methods that treat the product to be designed as an integral whole (Stone, Wood & Crawford, 2000). In the latter approach, there is no explicit decomposition of the product's design or the specifications the product is to meet. Conversely, a modular architecture for a physical product utilizes explicit mappings between subsets of the product's functional model and subsets of its physical structure (Stone, Wood & Crawford, 2000). A hypothetical design for a bed can provide a simple concrete example of this potentially abstract distinction. In an integral architecture, a designer would map the bed's top-level function (e.g., to provide a suitable place to rest) to the bed's unitary physical form (e.g., a horizontal surface that is large enough to lie on and is comfortable). In contrast, a designer developing a modular architecture would analyze the main functional goal of the product, decompose it in order to specify the subsets of product functionality that are necessary to attain it (e.g., keep the sleeper warm; provide elevated support for the head), and then map these lower-level functions to specific product components (e.g., blankets and pillow, respectively). Note that this distinction in approach is made at the level of the tasks executed and the information utilized by a designer during the design process rather than any necessary difference in the actual physical output of the process. The end result of an integral architectural approach can conceivably be the same as that of a modular one (e.g., both bed designs may utilize pillows and sheets). It is solely a question of whether the design process includes the explicit decomposition of both a product's structure and its function, and whether these hierarchical decompositions are utilized to drive other aspects of the design process, such as the management of design teams or product testing. The categorization of a product component as a 'module' is determined by the existence of a one-to-one mapping of that component to an aspect of the product's functionality, rather than any necessary physical attribute (Ulrich & Tung, 1991).

In the context of well-understood and simple physical artifacts such as beds, the distinction between integral and modular approaches can seem illusory, and rightly so. In realistic terms, the only part of a bed's design that is modular is the use of industry-wide standards for size (e.g., queen, king, twin). This allows consumers to use the mattress 'module' of one manufacturer with the sheets 'module' from another, and be reasonably sure that the fit will be satisfactory without having to physically test it. As the number and type of design parameters and quality criteria increases, and their networks of interactions become more complex, the distinction between modular and integral approaches becomes more crisp. In the field, modularity is generally more easily identified when multiple products share components between them, or when multiple designers collaborate on a product design by individually designing distinct components of the product that rely upon each other in some tangible manner. Still, it is more fruitful to view modularity in design architectures as a continuum, rather than a dichotomy with integral architectures at the other pole. That is, any given design of nontrivial complexity will exhibit a degree of modularity, although the modules may only be explicitly identified as such a posteriori to the design process.

Another possible argument against the importance of considering modularity in design is that any rational design process will necessarily include a decomposition of functionality and structure. If modularity is truly an inherent and obvious aspect of this process, it may not be worthy of serious investigation. This argument presumes that a rational design process must employ a predominantly analytic approach to design, which is an open question. A full discussion of design epistemology is outside of the scope of this thesis, however, and so this question will be left unanswered. The focus forthwith is on a basic definition of modular architecture, followed by an investigation into the application of modular architectural principles to the effective and efficient development of complex learning systems.

<u>Components of a Modular Architecture</u>

Modular architectures are created by partitioning an overall product design into several modules and articulating the interfaces between these modules (Baldwin & Clark, 1997). From the perspective of the information content of a design, modular architectures are created by defining distinct subsystems within an overall system, and then identifying the informational transactions required for the subsystems to operate together. The information flow across a module's boundary constitutes the interface between that module and the rest of the system.

Another way to consider the role of interfaces in a given design is to focus on the information that they explicitly do not contain. The total set of design parameters relevant to a module's operation can be divided into two categories: design parameters whose values are relevant only to that module, and design parameters whose values are relevant to other modules in the designed system. Returning to the example of a modular architecture for a bed, we might identify two parameters of the 'sheets' module: the material used for the sheets (e.g., cotton), and the dimensions of the sheets (e.g., queen size). Designers working on the sheets module would be relatively free to change the material used for the sheets without notifying design teams responsible for other components in the design, since the material used for the sheets would likely not affect the material used for the frame, for example. If, however, the designers in charge of the sheets module decided to change the dimensions of the sheets from queen to twin size, there would be implications for the design of the frame modules, and perhaps for other modules as well. In this case, the design parameter (and its value) would need to be explicitly specified and available to designers outside the module's design team. Given that an aspect of effective management of multiple design teams is the coordination of their separate design efforts, it follows that distinguishing design parameters that have system-wide significance from those that are module-specific is a critical task, since its output specifies the very portion of the design that must be coordinated from that which does not.

The mechanism for accomplishing this delineation is the module's interface. By identifying which information is visible to other modules, the interface also designates other design parameters as invisible or hidden (Baldwin & Clark, 1997). The design team working on a given module is free to vary hidden parameters without risking problems in component integration, and thus can tailor these aspects of the module's design to the objectives and constraints of the module's context. If new quality criteria emerge that affect previously hidden parameters, then it is the system architect's responsibility to communicate this information rapidly to all design teams implicated by it. For example, in the context of a bed's design, if is decided midway through the project that the bed's pillows will be fit into pillow-shaped depressions on the mattress, then the pillow designers will need to agree on pillow size with the mattress designers.

Given a set of possible designs wherein each design conforms to the requirements specified by the module's interface, designers will also require metrics to differentiate between them and determine which is preferable (Baldwin & Clark, 1997). Candidate designs for a module are therefore evaluated against two sets of quality criteria: global criteria that codify the module's responsibility to the larger system; and local criteria that frame the module under consideration as its own system. For example, in the bed example, all potential pillow designs under consideration must fit the pillow-shaped depressions in the mattress; once a set of designs that fulfill this objective are generated, a means for selecting one for production will need to be identified.

In order to derive the external quality criteria for a design, the product's functionality needs to be specified. The target functional goals of a modular architecture serve as an explicit definition of the set of broad functions that a product is to accomplish in order to satisfy the requirements of the larger system that the product will be implemented in. Essentially, these functional goals are analogous to Mohr's (1992) concept of 'outcomes of interest' in the evaluation of intervention programs. Outcomes of interest are defined by Mohr as those impacts of an intervention that can be traced directly to activities in a given intervention program and theoretically linked to the desired higher-level outcomes of the intervention. These outcomes of interest are considered to be inherently valued within the design of an intervention, in that evaluation stakeholders agree that their attainment constitutes satisfaction of all derived higher-level outcomes. In the context of product architecture, then, the attainment of a target set of functional goals can be considered synonymous with satisfying the full set of external objectives for the product.

In a complex product, the members of a set of target functional goals will tend to be untractable as drivers of a design process because they will be defined at a high level of abstraction. Therefore, the target functional goals will need to be hierarchically decomposed until the scope of each goal is sufficiently reduced to the point that it can be used to define the functional requirements of individual components in the product (Stone, Wood & Crawford, 2000). This information requirement provides a stopping rule for the decomposition process, in that the top-level functional goals are progressively decomposed until all subgoals are sufficiently small-grained so that each can be directly mapped to a specific module in the system to be designed. This information is used to drive fault diagnosis, allowing the tracing of failures in reaching a given functional goal to specific components in the product's design.

In order to test the validity of the functional decomposition, the role of each subgoal in satisfying its requirements in the hierarchy must be made explicit in such a way that metrics can be derived for evaluating product functionality in an incremental manner (i.e., testing of product subassemblies). These metrics are essential for the definition of testing processes that manage the two kinds of risks involved in modular architecture, namely component risk (i.e., whether an individual product module meets the functional requirements of its associated subgoal) and integration risk (i.e., whether an assembly of components produces a set of child functions which synthesize to accomplish a parent function). The specification of subgoals in this manner is also required for the analysis of the validity of the functional decomposition in and of itself.

The required elements in a modular architecture can therefore be summarized as:

- A set of target functional goals.
- A hierarchical decomposition of the set of target functional goals, such that any functional goal can be either mapped to the output of a specific module or to the additive product of two or more child modules functioning in concert. This decomposition also required associated metrics for the evaluation of functional subgoals.
- An overall product design partitioned into modules.
- Interfaces for each module which specify what the module must accomplish within the overall design, and metrics that articulate these global requirements in a manner that allows a module's satisfaction of these requirements to be verified.
- Specifications for local design parameters that can be hidden within each module, and metrics for determining the relative quality of a module design with respect to these design parameters.

Hierarchical Decomposition of Objectives as a Form of Modular Architecture

The classical instructional design technique of hierarchical decomposition of objectives (HDO) provides an example of the application of a modular architecture in terms of content development. One perspective on this technique is that, given a set of target performance objectives for an instructional intervention, the performance objectives are iteratively decomposed until any given objective falls into one of the following categories: it is a prerequisite knowledge/skill, it is the outcome of two or more child objectives, or it can be mapped to the outcome of a specific unit of instruction (e.g., Smith & Ragan, 1993). Although variants on this perspective exist, as in the application of the PxC algorithm (i.e., the probability of adequate performance without training multiplied by the cost to the organization of inadequate performance) (Patrick, Gregov, & Halliday, 2000) in place of the definition of prerequisite knowledge/skill as a stopping rule, the gist of the technique remains the same. A more precise expression of the HDO technique, however, might include the restriction that the network of performance objectives and subobjectives must be modeled successfully as a directed acyclic graph (DAG). In a DAG, directional links are employed to establish relationships between nodes such that any given node may have multiple parents (upstream) and children (downstream) but there is no path that eventually leads back to the node itself. This restriction captures the essence of the technique, which is the functional analysis of expertise into a well-structured framework through the process of hierarchical decomposition.

Similarities of the HDO technique to modular architectural approaches in physical product development include:

- A complex design goal is iteratively decomposed until only manageable design subgoals remain.
- The modules are "assembled" into a complete product (i.e., expertise) by virtue of each learner progressing up the hierarchy of objectives by interacting with a sequenced set of materials, in order to achieve the target set of objectives.
- The objectives for a module constitute its interface within the content architecture, that is, the target objectives of an instructional module represent its role within the overall content architecture.
- Individual instructional designers can be assigned to different modules, and can focus on meeting their module's objectives without worrying about implications for other modules.
- Operationalized performance objectives provide a mechanism for instructional designers to validate whether the instruction content of a given module meets the functional requirements of

that module in the content architecture. For example, by using three-component objectives that specify terminal behaviors, conditions of demonstration and performance criteria (Mager, 1962), instructional designers obtain a means for post-testing learners to determine if a content module is fulfilling its role in the system as a whole.

HDO can be considered as a hallmark of the 'systematic' school of instructional design. In an ideal execution of the technique, HDO should, at least in principle, inexorably lead to success – each and every necessary instructional objective is mapped to a distinct module of instruction designed to achieve that objective. In this ideal sense, if a learner failed to achieve the target set of objectives, faults in the design should be traceable as:

- Component failure, or the failure of a given module to fulfill its requirements as specified by its interface. In the context of an instructional design, this would mean that one of the content modules failed to catalyze its corresponding knowledge/skills objectives in learners.
- Integration failure, or the failure of two or more child modules to combine to provide a given parent module. In the context of an instructional design, this would mean that a parent objective was not attained through the synthesis of the designated set of child objectives.

For many target knowledge/skill goals, however, this ideal execution of HDO is difficult to achieve in practice. Expertise often does not neatly decompose into a finite set of discrete and additive knowledge and skill components the way a physical artifact such as a car does into engine, steering mechanism, chassis, etc., or even the way many software applications decompose relatively easily into modules that handle differentiable information processing functions. In this regard, HDO-based instructional design, when taken to its conceptual extreme, may share an unwieldy assumption with early work in artificial intelligence, namely that cognitive acts could be reproduced "through a 100 percent top-down approach: functions calling subfunctions calling subsubfunctions and so on, until it all bottomed out in some primitives. Thus intelligence was thought to be hierarchically decomposable..." (Hofstadter, 1985, 653). Luckily, humans are capable of graceful degradation in the face of incomplete information, and are also capable of independent and creative learning acts. It is therefore not necessary for a content architecture to contain every necessary bit of expertise that the execution of its target set of objectives would require.

Adept human learners can fill in the gaps on their own. Of course, there is a price to be paid for working in a design domain that does not enforce completeness of description; novice instructional designers often incompletely specify task models for the very reason that they cannot 'run' their model as an algorithmic process and check for errors (Patrick, Gregov & Halliday, 2000).

HDO remains a useful instructional design technique because it provides an approach or 'design move' (Schön, 1983) for translating an abstract set of target objectives into more tractable ones for the purposes of managing design complexity. This translation process is accomplished by modularizing the overall content architecture from the perspective of performance objectives. Thus, the field of instructional design can be said to already possess at least one instance of a modular architectural technique, albeit one whose role is primarily restricted to front-end analysis. Further incorporation and evolution of modular approaches in instructional design project management, however, may be stymied by certain assumptions inherent to 'systematic' instructional design in the broad. These are explored in the next section.

'Systematic' Instructional Design

'Systematic' is a term that has been associated with a large number of the models and methodologies proposed for the instructional design process in the second half of the past century (cf. Skinner, 1954; Gagne & Briggs, 1979; Carroll, 1990; Dick & Carey, 1990; Smith & Ragan, 1993; Jonassen, 1994). These models as a group represent an attempt to implement a highly disciplined analytical approach into the process of instructional design (Carroll, 1990). This analytical approach is commonly operationalized through an ordered set of tasks whose composition and sequence display a considerable degree of similarity across models. In a comparative analysis of 40 early models of instructional design proposed in the literature, Andrews & Goodson (1980) found the following tasks in common (the percentages represent the proportion of models that included that task):

1. A set of objectives stated in observable terms, with broad goals and detailed subgoals (100%)
2. Goals and subgoals expressed in terms of pre-test and post-test instruments (70%)
3. Goals and subgoals analyzed for types of skills/knowledge required (73%)
4. Goals and subgoals sequenced to facilitate learning (58%)
5. Learning population profiled (e.g., prior knowledge) (65%)

6. Instructional strategy formulated based on subject-matter analysis and/or learning population profile (85%)
7. Instructional media selected (60%)
8. Courseware developed (85%)
9. Courseware subjected to empirical evaluation, followed by diagnosis and revision of design (95%)
10. Materials/procedures for courseware implementation (e.g., installation, maintenance, periodic repair) developed (70%)

Thus, in general terms, an archetypal model of 'systematic' instructional design can be depicted as a phased approach whose generic form encompasses the following steps:

1. the design context is analyzed
2. a design is developed based on the results of the analysis
3. the design is operationalized (e.g., through the development of artifacts)
4. the design is evaluated against the objectives of the project, and, if necessary, revised based on information generated by the evaluation
5. the design is implemented into the target system

Analysis, design, production, evaluation and revision steps are included in virtually all instructional design models published in the 1960s, 1970s and 1980s (Gustafson & Branch, 1997b). Another way to view this shared emphasis is to identify the common call for instructional design objectives that are articulated through detailed specifications and accompanied by an evaluation mechanism (Reiser & Ely, 1997). This drive to implement a highly disciplined analytical approach to design is not unique to the domain of instruction design (Carroll, 1990). The sequence of analysis followed by synthesis (the production and implementation of artifacts) followed by evaluation as fundamental processes in design is found in many different domains, including engineering, industrial design, and town planning (Lawson, 1980). Although it is a common goal of design methodologists across domains, the 'rationalization' of design through this systematic phased process remains one that is difficult to achieve in practice (Jones, in Carroll, 1990).

It is of interest that the introduction of systematic analysis may not represent a terminal stage in the evolution of design in these varied domains, and if it is not, what directions a design domain may subsequently go in. Lawson (1980) provides a book-length discussion of the evolution of design domains and the progressive development of the processes that designers follow within these domains. According to him, the first period in the development of a design domain is "blacksmith design", typified by "constructional skill" juxtaposed with "theoretical ignorance" (Lawson, 1980, p.12). Blacksmith design reflects a tradition-based approach wherein craftspeople employ unformalized models through traditional patterns handed down from generation to generation. This approach to design "(relies) almost exclusively on intuitive methods, and design ability (is) widely held to be innate and largely unteachable" (Lawson, 1980, 2). Apprentice designers perceive design as a mystical ability, with native talent as the sole driver of expertise. In line with these perceptions, design education during this period only occasionally focuses on interim process deliverables such as product specifications or prototypes, and then solely to develop individual skills that are considered necessary for the practice of the given craft, such as drafting or surveying in architecture. The emphasis in design education is thus squarely on product rather than process, and methodology in terms of explicit and communicable design process models is not in evidence here.

In the context of instructional design, an example of blacksmith designers might be those individuals who feel that the development of effective print-based instruction is primarily governed by the hazily defined skill/talent of 'good writing'. Hackos (1994) notes some common beliefs within organizations at this early level of process maturity in the development of technical documentation:

- writers do not need training, as they already know how to 'write'
- the writing process continues until the writer 'gets it right' and/or until the deadline is reached; the characteristics of 'rightness', however, defy explicit definition
- documents produced by writers do not need to be reviewed or field tested, provided they are produced by 'good' writers
- design projects involving writing are not managed through time and cost estimates, because each project is unique and accurate estimates cannot be made with any useful degree of precision

The image of the lone blacksmith designer applies in the broader context of educational technology as well. Molnar (1971) characterizes the period of educational technology prior to the introduction and proliferation of systematic instructional design process models as akin to that of a cottage industry, wherein single technologists working in isolation carried out all the tasks related to an educational technology intervention based on their personal intuition and experience.

From this nascent stage, a transition to a more explicit process model in a design domain is made when designers - and, more importantly, their clients - require more control over design and its outputs (Lawson, 1980; Hackos, 1994). This tends to occur when the end product of a design domain significantly increases in value to the organization or, more broadly, to society as a whole. Another driver emerges when the technological component of the domain evolves such that the impact of designers' decisions is substantially increased in scale or scope. Calls for increased accuracy in estimation and a general emphasis on accountability are common at this point, which tends to coincide with the emergence of 'professional' designers and certification mechanisms (Lawson, 1980).

Evidence for this relationship between external requirements for increased control over design and the rise of process models can be seen in the historical evolution of formative evaluation in educational technology: "no sooner had motion pictures been adopted for educational use than the professional press began calling for quality control" (Cambre, 1981, p.4). By the late 1930's, systematic sequential methodologies for evaluation of educational audiovisual products were "routine in some circles" (Cambre, 1981, p. 9), including the articulation of explicit objectives and quality criteria. In a more contemporary context, the increasing importance of training in tight labour markets and knowledge intensive industries coupled with the increasing cost of computer-based training systems may serve to drive calls for increased accountability in instructional design.

From the perspective of design methodology, this transition requires that design processes become more open to inspection and evaluation. The scientific method is often espoused as a model to be emulated, because of the attractiveness of its perceived characteristics of openness, rigor and replication of process (Lawson, 1980). In the historical development of formative evaluation in educational technology, this is reflected in the formal application of psychometrics to the evaluation process, and the growth of psychologists' interest in evaluation activities as a form of applied research (Cambre, 1981).

From the perspective of design education, this transitory period leads to the emergence of textbooks, lectures, and other communication artifacts that embody a formalism of process (Lawson, 1980). The design process is now perceived as something that is communicable and therefore teachable. One can learn to be a designer, and the learning can at least partially occur within a defined and managed interval rather than solely in terms of a lifelong individualistic discovery process. In this regard, Gustafson & Branch (1997b) note the use of the Instructional Development Institute's detailed design model as the focal point of five- to seven-day instructional design workshops given to over 20000 teachers in the late 1960s and early 1970s. A comparable model, the Intraservice Procedures for Instructional Systems Design, was employed for training instructional designers in the American military. The ubiquity of "Dick and Carey" as the most widely-adopted introductory text in academia for teaching instructional design (Gustafson & Branch, 1997b) is another example of the standardization of method that is characteristic of this period.

Once explicit design methodologies begin to appear, they are often eagerly embraced by those interested in managing design, regardless of their actual utility. As noted previously, the driving force behind process formalization is "the problem of exercising...collective control over the designers' activities", rather than addressing the actual informational needs of designers themselves (Lawson, 1980, 19). This unfortunately signifies the arrival of 'process for the sake of process'. As a result, during this transition, many designers in the field correctly view process models as extreme simplifications of 'real design' that have little relevance to the actual design process. For example, the deliverables associated with these models are commonly considered as 'busy work' whose output is not taken seriously by designers or anyone else in their organization (Hackos, 1994). The almost complete lack of validation of these design models (Gustafson, in Richey 1997) reinforces the perception that these models offer nothing of value to validate.

A Case Study of Modular Architecture in Practice

The evolution of courseware development at India's National Institute of Information Technology (NIIT) provides an interesting example of the phases an organization passes through in evolving a mature culture of instructional design. Today, NIIT's Learning Solutions division is a modern "courseware factory" (Bharwaj, Chandran, O'Neal & Gibbons, 1998, p.101), employing over 500 professionals in the development process, and managing a library of over 650 educational software titles. In its earliest days, however, NIIT's designers developed instructional resources as handmade, customized products on an individual basis. When the growing scale of NIIT's development effort reduced the feasibility of this approach, it became necessary to control and orchestrate the efforts of the individual designers. Thus, "the model of growth for the newly-created instructional development organization seemed at first to be one of moving the individual craftsperson from the workshop to the assembly line" (Bharwaj, Chandran, O'Neal & Gibbons, 1998, 103, emphasis mine). Style guides and layers of management and quality control were used in an effort to standardize and control output across individuals. NIIT instructional designers, however, could still be considered to be working as 'artisans' at this stage, as they were using a COTS (commercial off-the-shelf) authoring environment to assemble courseware out of very low-level primitives, and thus making thousands of instructional, control and style issues during a typical design project. This resulted in high levels of variability across outputs from different teams, decreased productivity due to increased decision-making load, increased costs for quality control and revision, and decreased flexibility for modification if instructional needs changed (Bharwaj, Chandran, O'Neal & Gibbons, 1998). Furthermore, these problems increased with the design complexity of the products. Extensive project support was required from NIIT's information technology division, and the greatest degree of success within this approach was generally attained by instructional designers who were themselves proficient in programming skills.

The next stage in NIIT's evolution arrived with the introduction of logic templates (Bharwaj, Chandran, O'Neal & Gibbons, 1998). Logic templates standardized the data model and functionality associated with a given type of instructional content, such as a multiple-choice assessment item. The instructional designer's role was to select appropriate templates for the current design context. Although logic templates standardized a certain aspect of NIIT's design process, it did not touch issues of message design or instructional strategy, for which a separate corporate design culture was emerging (Bharwaj, Chandran, O'Neal & Gibbons, 1998). A further complication was that content authors were frequently making title-specific modifications to the logic templates. These modifications required that the associated template be tested again to ensure that it would perform both its original and new functions correctly, a nontrivial endeavour when a template had numerous conditions and branches. The costs and delay incurred from testing and revising modified templates often eliminated the original gains stemming from the use of a template-driven approach.

To remedy these problems, the NIIT developed a proprietary authoring system that operationalized the types of standards typically outlined in an organization's style guide, production manual or design bible. Any modifications that individual designers wanted to make to these standards would now have to be implemented in the authoring system itself, opening them up to scrutiny and review. By centralizing the mechanism by which modifications were introduced, the transfer of these mechanisms to other projects was also automated, which increased responsiveness to new design knowledge across teams during the course of a project. Another key aspect of this authoring system was that it separated and distinguished application logic (e.g., specification of navigation or branching control) from what was conventionally considered to be instructional 'content' (e.g., instructional text, animations, etc.). Although the logic templates used earlier by NIIT also embodied this separation, they were only used as such at author-time. Products developed using the new authoring system, however, actually manipulated elements in an internal database at run-time, thus establishing the carryover of a modular architectural approach to the actual structure and functionality of the learning system itself.

After developing titles using the new authoring system, instructional designers at NIIT observed that the potential existed to create generic drivers for specific types of instructional strategies. Another issue that emerged was a growing process bottleneck incurred during the translation of products to other languages. To further improve the instructional development process, NIIT evolved the authoring system into the so-called model-centered learning architecture (MCLA). In the MCLA, static instructional content is decomposed into objects that are stored in ODBC (open database connectivity)-compliant databases.

These databases are used to drive a layer of 'middleware' templates that are considered to be platform and format independent. These templates express the logic or functionality of the instructional content, and thereby capture its dynamic aspects. A third layer of drivers translates the logic templates to the context of a given delivery platform or format. This three-tier architecture means that the presentation, logic and static content facets of an instructional artifact are de-coupled from one another. Rather than being restricted to functioning within a single integral whole, each of these modules can function in any number of different product configurations.

There are several advantages of these modular architectures for learning system development, operation and maintenance:

- If an update needs to be made to a given static content object, the task only has to be performed once, after which the change will automatically propagate throughout the system. This benefit results from the fact that the logic templates do not themselves contain embedded static content, but only pointers to the database location of the content objects associated with them. This reduces the cost, time and quality control overhead associated with a significant amount of content maintenance tasks. Furthermore, since all deployed content titles are automatically the 'latest version', versioning issues are substantially reduced.
- Subject-matter experts can access content in the database for authoring and maintenance without requiring support from information technology professionals. Specialized editors (e.g., wizards) for connecting to the database can be provided to structure the content development process. This also allows architects to control which roles access which parts of the system, allowing for example authors of static content to perform maintenance tasks without accessing (and potentially introducing faults into) the logic templates or software code.
- Delivery cycle time is reduced, since aspects of the instructional design related to content dynamics can be captured in the middleware logic templates and reused. O'Neal (1996) estimates that this can reduce overall development time by 50%, because development and testing of the control logic of an instructional interaction typically consumes a relatively large amount of resources. The middleware layer also supports the transfer of pedagogical 'best practices' within an organization.

- The system architecture increases the reusability, interoperability and durability of both static and dynamic instructional content. For example, static content and instructional strategies are relatively insulated from volatility in delivery platforms. This reduces the risk of incurring increasing amounts of legacy content over time in large-scale content libraries. Content objects can also be reused across presentation contexts. For example, both conventional tutorials and just-in-time job-aids can share static content. Similarly, products can be localized by changing the content databases, without modifying the logic templates or platform drivers.
- Separate system partitions can be established between aspects of each layer which are open to customization for an individual product's context, and those which are closed. For example, content objects that are intended for standardized implementation (e.g., corporate logos, definitions of specialized terms) can be made available through read-only libraries. The evolution and extension of the logic template library can be explicitly managed in this fashion as well.

It is important to reiterate at this point that a modular architectural approach is fundamentally about how designers think about the design process. Thus, the full implementation of a modular approach to instructional design should change not only the structure and functionality of the resultant learning system, but also the process by which it is designed. In this regard, Gibbons, Bharwaj & Richards (1998) describe the application of a multi-tiered design approach to the process of developing problem-based instruction at the NIIT. In this approach, design and development tasks are categorized under one of three structures: content; event; or logic. Content tasks involve performance analysis, problem selection, problem data collection, and media resource production. Event tasks involve the definition of problem frameworks and the design of learner-system interactions. Logic tasks involve the development, integration and testing of logic templates, which link the outputs of event and content tasks in a computer-tractable manner. The tasks in these categories are conducted in a 'single-parse' manner, meaning that the three categories are concurrently engineered rather than sequentially phased. According to Gibbons, Bharwaj & Richards, the only media-specific task in this process is interaction design. Interaction design involves the design of screen layouts, interface controls, movement logic, and messages associated with modeling and practice activities in the problem-based instruction. Thus, this task functions to map generic problem demands to specific pedagogical features and capabilities of the medium or media employed. By placing an interface between the problem set and the interaction specification, Gibbons, Bharwaj & Richards are effectively insulating the problem set from obsolescence in the event of changes in the instructional strategies employed. The addition of the third tier, logic, allows for the formalization and management of the computer-side instantiation of the interface between the content and event tiers. The three-tier architecture thereby supports instructional designers in conceptualizing pedagogical strategies in generic terms, by providing conceptual partitions between design facets that de-couple them from one another.

Human Factors in Design Management: Going Beyond 'Systematicity'

Although the evolution of instructional design at NIIT ultimately terminates in a successful modular architectural development model, it also serves to indicate the fits and starts an organization encounters in transitioning out of an individualistic orientation of design as subjective craft. Success in evolving a design culture in this manner is by no means guaranteed for any given organization (Hackos, 1994). Since changes to workplace processes frequently result in productivity loss in the short term due to the disruption of embedded practices before any benefits are reaped, managers may conclude from the results of initial attempts at standardization and formalization of process that these goals are untenable in practical terms, and revert to a craftsperson approach. Even if the organization as a whole transitions successfully in this regard, a certain degree of friction and even attrition may be unavoidable, as designers who perceive process formalization as an artificial and overwhelming constraint on personal creativity resist the initiative and/or elect to leave the organization (Hackos, 1994).

It would be misleading, then, to frame the NIIT case study as a normative model of design culture evolution - that every design domain will inevitably undergo a linear progression from an initial rudimentary craft to an interim stage where the false precision of technical rationality is embraced (Schön, 1983), only to arrive finally at a more mature state of postmodern process enlightenment. It would be even more misleading to extend this argument and assume that all, or even most, individual designers who could be considered as experts in their domain pass through the same stages, in an 'ontogyny mimics phylogeny'-style manner (although some support for this line of argumentation might be obtained from Perry's (1970) scheme of intellectual and ethical development). In examining research on instructional design process management, it can seem that for every opinion leader in educational technology who calls for increased 'maturity' in the field through process formalization (e.g., Lesgold, 1998), there are those who believe that the design studio in the fine arts is the proper model for instructional development (e.g., Orey, Rieber, King & Matzko, 2000), or at least that personal criteria cannot and should not be excised from determinations of quality in engineering criteria in complex learning systems (Winne, 1993).

Thus we arrive at the human factors of design management, since instructional design processes, despite their possible formalization through structured models and their instantiation in detailed deliverables and electronic performance support systems, are in the final analysis still enacted by human designers. It must be accepted that design as a human activity is complex and unwieldy in and of itself, and the manner of its execution resists easy optimization. This is not to say that this point necessarily needs to be accepted on an epistemological level. That is, agreement that real-world design projects are generally underspecified (e.g., Kolodner & Leake, 1996) does not necessarily imply that this point need be accepted as a philosophical position in the sense of postmodern scientific concepts such as underdetermination and indeterminacy (Miller & Fredericks, 1991). Whether there is an a priori inability to specify a logically unique theory as true on the basis of a given body of empirical evidence is an open question. In the context of managing design projects, no claim need be made at this level. It is only argued that design projects of non-trivial complexity are typified by designers' inability to specify a unique optimal design based on the information at any given point in the project, in an analogous manner to specifying a line of best fit to a set of data points (cf. Garrison, 1986). Thus, for the same reason that formal proofs of correctness have limited utility in the evaluation of intelligent tutoring systems – the functions in these systems are incompletely specified and are therefore analytically intractable (Mark & Greer, 1993) – instructional designs as a whole cannot be 'proved' or 'disproved' in toto.

This does not, however, mean a de facto acceptance of a position such as that taken by radical constructivism (von Glasersfeld, 1995) within instructional design. If it did, then the future of educational technology may well be limited to faddish swings between absolutism and relativism, nullifying the possibility of real progress.

Lawson (1980) notes that a typical reaction to the perceived failure of the scientific qua analytical approach in dealing with design complexity is the assertion of a more utilitarian philosophy of practice. In this period, design expertise, while not perceived as an innate talent, is considered as something that is learnt rather than taught. Each designer must develop an approach to design for themselves, although this can be based on the study of other designers' works as well as developed through discussion with them. The design studio approach employed by Orey, Rieber, King & Matzko (2000) to train novice instructional designers is an example of this perspective. Here, an emphasis is placed on developing competency with production tools (e.g., designers' initial projects are not related to instructional design per se, but focus instead on the production of media objects), allowing individual designers to work on their own schedules and follow their own interests, and having them engage in discussions with peers as well as a team composed of several instructors. Arguably, many students and practitioners alike would respond warmly to such an approach, since it exemplifies the aspects of instructional design that are shared with domains that prioritize the development of a subjective aesthetic sense. It may also represent a constructivistic approach to instructional design being turned in on itself.

The danger here lies in damning the whole enterprise of design management in educational technology, and with it, more 'technical' approaches such as three-tiered modular architectures, on the basis of the initial flawed and simplistic approaches attempted under the guise of extreme technical rationality. Similarly, the possibility of improving management of design needs to be considered separately from the tensions inherent in the interaction between managers and practitioners in any field. Process formalization and standardization is often viewed by both sides here as a means by which to increase the accountability of design functions within an organization, allowing techniques for auditing the effectiveness of design processes to displace a 'trust' in designers' competencies, with predictable polarization as a result (Armstrong & Tomes, 1996). Increases in the ability to link educational technology interventions to positive impacts on workplace performance and an organization's bottom line can be of great benefit to its practitioners, however, because it can free them from the 'dead' budgets these projects are generally accorded; in the absence of a language linking actions with impact, design management tends to focus on development expense and delivery time at the expense of performance factors (Reinertsen, 1997). Thus, the limitation of measurability in design processes to cost-side metrics leads to management's facile equation of efficiency with economy in this regard. Rectifying this situation requires that design in educational technology move beyond the simplifications of its classical systematic period. The innovation of design processes will occur in a significant sense only if a focus is placed on meeting the actual informational needs of designers in the field.

With respect to the human factors of process innovation, then, this might mean attending to such seemingly trivial details as the graphical portrayal of process models. For example, classical systematic instructional design models may be interpreted by practitioners as "stifling, passive, lock-step and simple" (Gustafson & Branch, 1997b, p. 78) as a direct result of the visual elements generally used to represent these models: rectilinear rows of boxes connected by one-way arrows, with the return line in the process parallel to the straight main line. In their place, Gustafson & Branch advocate a more realistic curvilinear portrayal employing ovals joined by curved lines with two-way arrows, which better symbolize the iterative cycles that occur during design. Of course, merely adopting a new representational scheme for design processes is superficial if change is restricted to this level alone. In a similar vein, Reiser & Ely (1997) note that, in a recent definition of instructional technology published by the Association for Educational Communications & Technology (Seels & Richey, 1994), the word systematic is specifically excluded "to reflect current interests in alternative design methodologies such as constructivistic approaches" (p.69), despite the fact that the ensuing definition retains essentially the same features as the models originally proposed in the 1960s and 1970s. Again, true innovation requires more than a mere cosmetic facelift of nomenclature and symbol.

For more meaningful progress, new conceptions for design models must embrace the rhetorical facet of design and provide a framework within which designers can develop procedural sophistication. A notable example here is the proposal of Hannafin, Hannafin, Land, & Oliver (1997) for what they call a 'grounded practice' design model for the development of constructivist learning environments. Although the focus here is still on systematic implementation of processes and procedures rooted on established theory and research on human learning, an emphasis is also placed on identifying heuristics that help to deal with design complexity rather than searching solely for algorithmic prescriptions. Within grounded practice, design rationales are conceptualized as explicit, public, defensible theoretical frameworks that are operationalizable for validation and evolution. Hannafin, Hannafin, Land, & Oliver explicitly differentiate their model from those based on personal preference, pragmatic concerns, designers' previous experience with 'what works', and familiarity. In doing so, they reinvigorate design management by exploring the interstitial region between absolutism and relativism. In this context, systematic processes do not have to be 'correct' or 'incorrect', and time-variant relationships can be accepted within them (cf. Jonsen & Toulmin, 1988). Hopefully, then, the rumours of the death of systematicity in instructional design may be somewhat exaggerated; the coffin may well turn out to be a chrysalis, the epitaph an epiphany.

In the next section of the thesis, the details of a modular architecture for assessment systems are explored in the context of providing a formalized, explicit, and operationalized framework for instructional design.

Evaluation of the IMS Question & Test Interoperability Information Model

Distributed Assessment Systems

Introduction

Before engaging in a discussion of distributed assessment systems, the concept of a distributed learning system in the broad must first be briefly outlined. An example of such a system is that proposed by Wu & Tam (1998), which describes a client-server network involving multiple servers in the content development and delivery process. In this system, individual authors develop instructional content and make it available on the World Wide Web through authoring servers. Each content unit so published has a Courseware Information template associated with it, containing data describing the content in a structured manner in terms of such aspects as its subject matter, delivery medium, and prerequisites. The data template also contains a Uniform Resource Locator (URL) for the content, so that it can be located and retrieved through the network. Over time, the amassed efforts of numerous authors creates a distributed library of content topics, distributed in the sense that it exists on numerous servers connected through the WWW. In order to search this body of content for a given educational need, individual learners access it from client software connected to the authoring servers through a courseware server. Their interaction with the courseware server is also structured through two prefabricated templates, called the Curriculum template and the Teaching-Style template. Although operational details are lacking for these data models, presumably they structure the framework within which learners articulate their educational goals, as well as the manner in which the courseware server configures and delivers the retrieved content.

Architectures such as that described by Wu & Tam (1998) share certain common elements:

- The conventional concept of courseware as a single custom-built software application with embedded content on a given subject is supplanted by the concept of a library of content stored on multiple distributed servers linked together by a network and accessed through client applications. Content is thereby separated from the software that utilizes it, and there are often several distinct layers of applications involved in the operation of the learning system.
- In many cases, the distributed learning system is envisioned as a relatively open system, with multiple authors and/or organizations producing content for a broad audience of learners. Even in cases where the distributed learning system is a proprietary one developed for internal use by a single organization, the system is still open in a temporal sense. Content is continually added to the system during its lifecycle, and the system architecture and infrastructure will likely undergo major evolutions rather than being limited to revision through maintenance releases.
- This architectural shift also impacts the approach taken in analyzing and assessing these systems. Rather than focusing primarily on individual courseware titles, produced within an 'author once, use many' context, distributed educational systems are also diachronically evaluated at the system level, with numerous authors creating numerous content components over a long period of time. Thus, the more traditional unit of analysis reflected in hypotheses of the form "what is the effect of computer-based tutorial X on developing learner's expertise in domain Y" is joined by investigations into the effectiveness and efficiency of large collections of relatively diverse content. This is paralleled in a change in management focus, as tactical 'go / no-go'-style decision-making about individual training products is joined by strategic decision-making concerning system-level investment initiatives and organizational knowledge management.
- The identification of content for a given learning need is a primary design concern for these systems. The efficiency and effectiveness of search techniques based on browsing or scanning the content itself

will be increasingly reduced as the size of the information space increases (Marchionini, 1995). Another aspect of the content identification issue is that open systems with multiple authors will generally contain content with high variability on such design factors as breadth of topical coverage, difficulty level, and pedagogical approach. As a partial response to these two issues, distributed educational systems often utilize metadata schemes for structuring the manner in which content is represented for the purposes of search and retrieval. The metadata scheme also implicitly or explicitly serves as a reification of part of the content design space, lending a certain degree of structure to the authoring process. For example, through the use of a metadata scheme that includes the classification of educational content on the basis of the average time required to learn the content, content authors are made aware that this is an instructional design facet that has significant relevance within the learning system.

- Another primary design concern is content granularity. Content granularity refers to the relative size of the content used in a distributed system, in terms of capturing the difference between a brief explanation of a given topic and a complete lesson or entire course on it (Wiley, South, Bassett, Nelson, Seawright, Peterson, & Monson, 1999). This becomes an issue once simple retrieval of individual, integral content files is replaced by the more complex process of content configuration. That is, rather than addressing a given learning need through the single content file that best matches it (i.e., one-to-one mapping of content to need), learning systems provide mechanisms for combining multiple content files into a package that provides the desired breadth and depth of coverage (i.e., many-to-one mapping of content to need). The addition of this type of functionality drives the granularity of authored content downwards, since maximal flexibility in systems of this nature is predicated in part on the existence of a critical mass of small-grained content that can form the basis of multiple reconfigurations. In these systems, content authors are often said to produce content 'objects' rather than complete 'lessons' (e.g., Murray, 1998; Wieseler, 1999), the latter arguably being the primary grain size of authoring for traditional classroom-based delivery. In part, the recent prevalence of the 'object' term in the educational technology literature may be attributable to the growing influence of software engineering on instructional design processes within complex computer-based environments (e.g., Tripp & Bichelmeyer, 1990), especially in terms of object-oriented analysis and programming (e.g., Yourdon, 1994) in instructional content development (e.g., Chapman, 1994). A more central reason, however, for the adoption of 'object' as a term in instructional development is that it is a semantically neutral term with respect to the granularity of instructional content, unlike 'topic' or 'lesson'. Regardless of whether the specific term is 'knowledge object' (Merrill & ID2 Research Team, 1996), 'learning object' (Wiley, in press), 'educational object' (Graves, 1999) or 'information object' (Wieseler, 1999), 'object' reinforces the notion of a design space wherein the basal content units themselves are open to definition. This type of design approach accepts the existence of multiple possible configurations of content and variable structural forms, and makes the definition of these structures an explicit facet in the design of a learning system's architecture.

- For the purposes of simplification, distributed assessment systems are discussed in this thesis as standalone distributed systems, that is, in terms of their operation in isolation from other types of distributed systems. It should be borne in mind that assessment systems generally function as components within larger systems that also perform other services, such as instructional content delivery and/or knowledge management. Although outside the scope of this document, an important aspect of distributed systems in general is their connections to larger systems of which they are subsystems, as well as their interactions with other systems with which they exchange data.

In addition to the concepts described above which arise due to the 'distributed' aspect of distributed evaluation systems, there are also concepts related to the 'evaluation' aspect of distributed evaluation systems, particularly in terms of the implementation of evaluation instruments in a computer-based environment. A reasonable starting point from which to begin investigating this aspect can be obtained by defining a simple case of a computer-based assessment system, one that exhibits limited complexity in terms of functionality and structure. Such a case might be an evaluation system that is solely responsible for displaying a set of static items in a single fixed linear sequence, and recording the responses made by the candidate. The closed set of items would be stored within the assessment application and would be of similar type, for example a series of multiple-choice questions. The candidate would enter a response for each item, and the assessment system's responsibility for the response process would be limited to providing a mechanism for response entry, such as recording keystrokes, and associating the keystrokes with the item currently displayed. After the assessment session was complete, the transcript of the session would be manually scored by a human scorer. In this context, the automated role served by the computer-based system is relatively minimal, and the advantages of this form of system over traditional paper-based assessments is likely minimal as well.

Relative to this functional baseline, the automated evaluation of items represents a new level of functional sophistication. Simple implementations of this feature would only require that the system be capable of storing the correct response to each item as data, comparing the responses entered by respondents against this data, and incrementing a variable to represent the accumulated number of correct responses. At the macro level, these simple evaluation systems might also be capable of tracking the duration of the assessment session, and limiting it to a prescribed interval.

If the context of the assessment system includes learners engaging in assessment for the purposes of practicing newly-acquired knowledge or skills, the addition of functionality for providing feedback becomes desirable. Basic examples of feedback schemes (Mory, 1992; Clark, & Dwyer, 1998) include:

- Knowledge of response (KR) feedback, where learners are informed as to whether each individual response is correct or not
- Knowledge of correct response (KCR) feedback, which adds the identification of the correct answer when an incorrect answer is made
- Elaborative feedback, which includes an explanation of why the correct response is correct and/or why the incorrect response is incorrect

More sophisticated feedback schemes attempt to exploit the information available from response patterns (i.e., sets of responses across items) to select appropriate feedback events. Computer-based environments that attempt to model the learner's knowledge state and utilize this model in automated pedagogical decision-making, such as intelligent tutoring systems (Sleeman & Brown, 1982), represent a further extension of this concept. In the case of tutoring systems that mimic Socratic dialogue-style tutorial interactions, the concept of assessment as a distinct entity from instruction eventually disappears, as assessment and feedback becomes a continual feature of the instructional process, interwoven with exposition and demonstration at the micro level.

Feedback selection represents an instance of a larger class of functionality that falls under the rubric of the adaptive system. An adaptive system is by definition a dynamic one that has the potential for changing its course of action based on its interaction history with a given user (Tennyson & Park, 1984). Thus, these systems extend consideration of aptitude-treatment interactions in instructional design to state as well as trait variables (i.e., to implementing variations in instructional delivery at run-time as well as at author-time), as well as to other variables beyond learners' aptitudes. Although the feedback schemes described above are a common expression of an adaptive system's functionality, adaptive systems are by no means limited in their scope of adaptation to this design facet. Another important form of adaptation is demonstrated by the category of computer-based assessment known as computer adaptive testing (CAT) (Weiss & Kingsbury, 1984). In CAT, items are selected for respondents on the basis of their performance on past items. Although the operational details of CAT systems differ, the basic principles involved can be illustrated by the progressive approximation of ability estimates employed in the maximum information approach (Rudner, 1998). In this form of CAT, each item in the CAT's item bank has an item response function associated with it, which describes the probability that a learner with a given ability level will answer the item correctly. This item response function is defined using item response theory, such as the three-parameter item response model (Birnbaum, 1968) which ascribes a numeric representation for an item's difficulty, discrimination, and the probability that a respondent with extremely low ability will obtain the correct response. The information function for an item is defined by the standardized slope of the item response function at each level of ability. At any given point in the CAT session, the current ability estimate for the respondent is used to identify the item with the maximal value for its information function at that ability level, and this item is selected for administration. The new information generated by the respondent's response to this item is used to revise the ability estimate until a terminating condition of some form is attained.

Perhaps the most technically sophisticated form of computer-based adaptive assessment currently in operation joins adaptive testing with automated item generation (Embertson, 1999). In these systems, the underlying factors that affect item response parameters are empirically derived, and then used to devise algorithms capable of generating items at run-time with the desired informational properties. Although these systems are only feasible in domains where content can be generated algorithmically, they represent an exciting frontier for research nonetheless.

A parallel line of development with respect to the increasing sophistication of computer-based assessment systems lies in their ability to handle diverse item and media types. With the arrival of interactive multimedia at the desktop workstation, assessment items in a computer-based environment are no longer limited to simple true-false, multiple-choice and fill-in-the-blank formats. For example, high-fidelity simulations of real-world environments provide a mechanism for the assessment of knowledge and skills in a manner beyond that feasible with print-based media or in the classroom. As with intelligent tutoring systems, the new possibilities afforded by computer-based educational environments in terms of media handling result in a qualitative shift in the operational concepts of assessment.

It is the union of the concepts described above which forms the definition of the distributed object-based assessment system. Although a potentially powerful combination, the 'open' nature of distributed systems coupled with the need for well-structured data in the case of computer-based automated processing creates a problem. In order for nodes to exchange information within a networked system, a communication protocol must be agreed upon. This is the primary driver for the recent development of standardized data models for distributed learning environments, of which the IMS Question & Test Interoperability specification (Smythe & Shephed, 2000) is an example.

Interoperability and Reusability

The primary objective of the IMS Question & Test Interoperability specification (hereinafter, QTI) is to describe the data structures necessary for interoperable content in the context of computer-based assessment, with a primary focus on distributed systems operating over the Internet (Smythe & Shephed, 2000). The distinction between interoperability and related quality criteria such as reusability can be fuzzy at times, as it is dependent on the type of content unit and the context of the system or systems that the content unit functions within. In this thesis, interoperability is defined as the ability of a given component to function in the same manner on different automated systems, without the intervention of humans. In the context of assessment content, interoperability in its fullest sense would require that the content should exhibit identical structure and functionality when imported into any assessment system that was compliant with the data standard, without requiring additional development or customization work. Interoperability is therefore a technical quality criterion, and a subset of the broader criterion of reusability, which also focuses on the ability to use a component in more than one system, but allows for the necessity of human intervention in this process. Since the potential value of an interoperability standard such as the QTI specification may also be derived from increased reusability of assessment content, it is important to make the contexts in which this is expected to happen explicit:

- The separation of the logic of assessment content from any specific authoring or delivery system protects this information from changes in these systems. For example, an organization that migrates to a new delivery system will not have to revise or redevelop existing assessment content, provided that the content and the new delivery system are compliant with the QTI data model. In this sense, a given set of QTI-compliant assessment content is reusable across different instantiations of QTI-compatible authoring and delivery systems, because it is interoperable between them.
- QTI-compliant delivery systems will be responsible for delivering assessment content in an appropriate configuration for the delivery platforms they are compatible with. For example, a QTI-compliant delivery system that can accommodate both desktop workstations and handheld PDAs will likely utilize different interfaces for the assessment content in these two cases. By separating the data and logic of the assessment content from its instantiation in a given interface (i.e., presentation), this information can be reused, as it remains constant between the two platforms. Here, the combination of the QTI data model and the delivery system (or systems) makes the assessment content interoperable between delivery platforms. Note that the definition of assessment content explicitly includes both the assessment logic and the data processed by the logic.
- The previous two examples illustrate contexts where assessment content is seen as an integral whole, and the dynamic elements are authoring / delivery systems / platforms. Another potential advantage of standardized data models such as the QTI are that assessment content from different sources can be retrieved and configured according to the needs of a given context. For example, authors of assessment

instruments may search a network for specific types of assessment items from multiple distributed item banks. In principle, the search process is made much more efficient if the items are coded through standardized metadata, which the QTI data model provides for. Furthermore, the authors can be confident that QTI-compliant items will seamlessly integrate into a single assessment instrument, since they share a single method for representing the item's logic and data. The QTI data model also provides mechanisms for representing the logic and data associated at the macro level of an assessment instrument, such as the procedures for selecting and sequencing items within the instrument. In this sense, data standards enable content reuse at the micro level, since individual assessment items are interoperable between assessment instruments.

- A final aspect of reuse focuses on the architecture of an assessment system. By adopting a data model such as the QTI, an organization obtains an explicit specification for the structure and functionality of a computer-based assessment system. These core elements of the assessment system's architecture can serve as a common module for other aspects of assessment development and implementation, and can be managed as an explicit entity by a limited number of architects. Compared to development models wherein core architectural elements are open to modification by all courseware developers, the ability to control evolution of the system architecture in a positive manner is enhanced because it is possible to separate facets of assessment design and development into open and closed categories.

It is important to note that some of the aforementioned potential benefits rely on the data standard being a non-proprietary one. This is what differentiates an open standard such as the QTI specification from the data model used internally in a learning management system such as the commercial WebCT software (WebCT, 2000), for example. Assessments authored in a proprietary data format such as that employed by WebCT cannot be ported to a different delivery system, and therefore become legacy content if an organization decides to migrate away from the system associated with the proprietary data format. Conversely, a primary reason for adopting an interoperable data standard is to protect investments in content development from risks associated with changes in the assessment system's infrastructure.

In addition to the potential benefits discussed earlier that arise from interoperability and reusability, adoption of the QTI data model may also have other positive impacts:

- By explicitly separating assessment content into data, logic and presentation components, the QTI data model can provide a structured conceptual framework that instructional designers can use to analyze their designs. In addition to providing an explicit 'design space' for experienced designers to work within, the QTI data model can provide a tangible representation of assessment content for novice instructional designers. For example, the QTI data model could be used in introductory classes on assessment design, even without the model being implemented in a functioning system.
- By identifying the components of an assessment instrument, the QTI data model can serve as a basis for deriving the responsibilities of these components from different perspectives on the assessment system, facilitating communication within multidisciplinary content development teams. A notable example here would be in the case of a team composed of instructional designers and multimedia software developers. Even if the team was not developing strictly QTI-compliant assessment content, the QTI data model might serve as a highly fleshed-out starting point for understanding the information requirements of each role, and developing deliverables that met these requirements. At the very least, the QTI data model could be used in defining the functional scope of the planned assessment system, in that the components of the QTI model that lay within the scope could be identified. The QTI data model's components could then be used as a starting point for deriving the functional specifications for the corresponding components in the system under design.
- By providing an explicit operationalization of logical constructs in assessment content, researchers' abilities to empirically establish the relationships between these constructs and other variables of interest, such as item difficulty, may be increased. Research on these relationships has been hampered in the past by a lack of explicit definition of these constructs (Anderson, in Roid & Haladyna, 1982).
- By formalizing the logic and data used in assessment instruments within an organization, the QTI data model may allow for better measurement of progress in content development projects, and higher levels of quality control and risk management. These benefits arise from the fact that QTI data model provides an explicit structure upon which to formalize aspects of the content development process.

In evaluating the potential benefits of adoption of the QTI data model, one must also bear in mind that the specification may suffer from the same potential drawbacks as any data standardization effort:

- Components of the specification may be underdefined, or there may be inconsistencies or contradictions between components. Processing of standard-compliant data may therefore not function as envisioned.
- The functional scope of the data specification may be so wide that it is cost-prohibitive to develop compliant systems.
- The functional scope of the data specification may be so narrow that it does not cover an adequate number of item types.
- The specification itself may be too complex, raising the amount of training and effort required for the individuals who will have to work with it to a prohibitive level.
- The formalization of a design process, in and of itself, may be rejected by individual designers.
- The organization adopting the specification may not be mature enough with respect to process formalization to provide the necessary infrastructure or culture for standards-based design methods.
- The limited scope of the organization's development and maintenance may not justify investment in the standard and the necessary change management.

A final issue that could undermine the success of a standard such as the QTI specification is that even if it achieves its primary goal of enabling the development of interoperable assessment content, no such content may end up being produced. In the vernacular, "if you build it, will they come?" That is, the existence of an interoperability standard may be necessary for open, distributed learning environments to emerge, but it is by no means sufficient for them to do so. There are considerable arguments both for and against the proliferation of distributed assessment systems. In any event, it is difficult to foresee the future, and the question has a certain "chicken & egg"edness to it that implies that the unit or level of analysis is incorrect. A more tractable line of inquiry focuses on the factors that would increase the value to an organization of adopting a standardized data model like the QTI specification:

- The larger the scale of assessment development and maintenance in the organization, the greater the potential value of a standardized data model for assessment content. For example, a company that maintains a library of 500 content titles will likely derive greater value from such a standard than one that maintains a library of 5.

- The greater the variety in delivery contexts the assessment content must serve in, the greater the value of a standardized data model that separates data and logic from presentation aspects. Therefore, a multinational corporation that maintains assessment content intended for delivery over the WWW to desktop workstations as well as PDAs will likely derive greater value from such a standard than a small local business that has a single computer dedicated for training delivery.
- Organizations that are more mature in terms of their implementation of formalized processes, especially with respect to instructional design and knowledge management, will likely derive more benefits from a standardized data model than organizations which develop content using primarily ad-hoc methods.
- Organizations employing development teams for assessment content, especially multidisciplinary teams, will likely derive more benefits from a standardized data model than organizations that assign individual designers to projects as integral wholes.

Despite the issues raised above, the quality of the QTI data model can still be productively evaluated at a lower level in terms of its effective separation of assessment content into data, logic and presentation elements for the purposes of increased interoperability. In the next section of the thesis, the QTI specification's degree of success in this regard is investigated.

Methodology for Evaluating a Data Standard for Computer-Based Assessment

The proper method of vetting data standards intended for distributed educational systems is open to discussion. The primary challenge here is that these data standards are intended for use in systems that for the most part do not exist as yet, because they require the establishment of a communication protocol in some form a priori to their development. In many ways, the situation is analogous to the early days of Hypertext Markup Language (HTML) specification development. It would likely have been a difficult proposition at best to predict the current manifestation of the World Wide Web (WWW) from the first version of the HTML specification. A complete evaluation of the standard at that time would have required not only that, but also a definition of the counterfactual set of possible forms the WWW might take. Despite this formidable analytical challenge, the need remains to at least partially assess a fledging data standard early in its development, when the costs of revision are substantially lower.

The approach taken in this thesis is that data standards for interoperability within a given domain possess both an internal coherence and an external scope. Thus, a standard might be found to have flaws if it was internally inconsistent in its structural or functional specifications, and it might also be found to have shortcomings in terms of its ability to address the full scope of situations it was intended to cover. An interesting wrinkle in the latter respect is that the specifications currently being developed under the auspices of the IMS Project are explicitly considered to be pedagogically neutral (Anderson, 2000). This is a common position taken in most of the recent standards initiatives in the domain of distributed learning systems (e.g., Farrance & Tonkel, 1999; Advanced Distributed Learning Initiative, 2000), possibly in the hopes that the data models can be developed and evolved to a reasonable extent on a technical level before being subject to the more abstract criticism involved at the pedagogical; that is, to optimize internal coherence before considering external scope. From one perspective, it could be proffered that a data standard for instructional content cannot be pedagogically neutral, since a standard inevitably constrains its adopters by virtue of its fundamental assumptions and inclusion/exclusion criteria. For example, it could be argued that the QTI data model, by focusing on standardized assessments, de-emphasizes constructivistic pedagogical models. This may be an unnecessarily high standard to hold a standard to, so to speak. As with an alternative hypothesis, the scope of a standard cannot be 'proved' or definitively established, since the existence of a single hypothetical case that it cannot accommodate for implies that its coverage is less than absolute, and such a case may always arise in the uncertain future. At a more tractable level for discussion, however, the QTI data model can be inspected in terms of its breadth of coverage of common assessment contexts. This preserves the ability to analyze the pedagogical implications of the data model alongside with the technical. Therefore, in this thesis, the QTI data model's external scope is assessed in terms of the degree to which it is pedagogically inclusive rather than whether it is pedagogically neutral or not.

In order to assess the pedagogical inclusivity of a data standard for computer-based assessment, it is first necessary to delineate a field of common forms of assessment against which to judge the standard's scope. Performance assessment can take a myriad of forms, including traditional print-based multiple-choice tests, oral examinations, athletic competitions, artistic portfolios, and self-evaluation. It would be unreasonable, however, to expect the first version of a data standard to encompass functionality that was not already present in existing custom-developed computer-based applications. As many computer-based assessment applications are conversions of print-based instruments, the first part of the task is to identify common forms of test items on these instruments. These forms include true-false, multiple-choice, matching, ordering, sentence completion, short-answer, CLOZE-procedure and essay formats, and were identified from texts on assessment development (Roid & Haladyna, 1982; Ebel & Frisbie, 1986; Linden & Hambleton, 1997; Haladyna, 1997; Osterlind, 1998).

An important byproduct of defining the design space of assessment forms is that it allows for a lexicon of test design terminology to be defined, which is necessary given that no single definitive lexicon has been established for print-based assessment (Anderson, in Roid & Halydyna, 1982; Osterlind, 1998). New forms of assessment arising from interactive multimedia learning environments further complicate matters in this regard. For example, a relatively fundamental distinction in print-based test items exists between 'selected response' and 'constructed response' items (e.g., Haladyna, 1997; Osterlind, 1998). In a selected response item, such as a conventional multiple-choice question, the respondent selects one or more responses from a list of options provided within the item. Conversely, in a constructed response item, such as a conventional fill-in-the-blank question, the respondent constructs a response using the elements of a given vocabulary (e.g., English). Thus, the distinction here lies in whether the respondent's task is to recognize a correct response from a small set of options presented as part of the item, or to recall a correct response and construct it from a very large set of elements that are unspecified in the item. In print-based assessment instruments, this distinction is a relatively crisp and useful one, but in the more flexible medium provided by computer-based environments, it loses a certain amount of power because of its lack of precision. In a computer-based simulation involving electrical circuits, for example, the respondent may be given a small set of electrical components and directed to construct a circuit with a given total resistance. Since the small, prescribed set of elements can be combined in a large set of possible configurations, this form of computer-based item shares some properties with both selected and constructed response items. Therefore, it is likely that the design space for print-based items will require a certain degree of translation and elaboration when applied to computer-based items.

Below, a survey of common assessment item forms identified in the literature is presented. After the presentation of the QTI data model, these item forms are used as a baseline against which the scope of this data model is evaluated.

Survey of Common Assessment Item Forms

Multiple-Choice Items

To begin the survey of common item forms, a reasonable place to start is with the ubiquitous multiple-choice exam question. As defined by Haladyna (1997), the conventional multiple-choice item consists of a question in the form of an interrogative sentence followed by three to five answer options, where only one of the answer options is correct.

Example:

Which of the following is least affected by outliers? Select the best single answer from the following list:

A). Range B). Mean C). Mode D). Median

This basic type of assessment item provides a starting point for establishing a terminology for computer-based assessment. The part of the item that defines the context for the response is labeled the 'item stem' (Haladyna, 1997; Osterland, 1998). Although commonly phrased as a question, it is not limited to presentation in this form. The specifications for selecting a response (e.g., "Select the best single answer from the following list") are the 'directions' (Osterland, 1998). The complete set of answer options is the set of 'response alternatives' (Osterland, 1998), with the correct answer being the 'correct response' and the incorrect answers being the 'distractors' (Haladyna, 1997; Osterland, 1998). Multiple-choice items with only two response alternatives (i.e., a correct response and a single distractor) are labeled as 'alternate-choice' (Haladyna, 1997). In order to refer to the individual making the responses, the term 'respondent' will be used.

Rating Scale Items

A variant on the multiple-choice format, which is somewhat more common in psychological testing than in educational evaluation, is the rating scale. In this item type, response alternatives are ordered polytomous categories (Samejima, 1997).

Example:

For each of the following items, enter the number which best reflects the extent to which the problem bothered you in the last week: 1 = extremely, 2 = quite a bit, 3 = moderately, 4 = a little bit, and 5 = not at all.

___ Thoughts of failing my statistics course

___ Feeling that I will never use statistical analysis in my professional life

___ Losing interest in studying statistics

This problem type differs in two notable ways from the conventional multiple-choice problem: firstly, it utilizes a set of response alternatives that are on ordinal or interval scale of measurement; and secondly, it often utilizes the same set of response alternatives for a large number of items. A final difference is that the scoring model for the collection of test items may meaningfully utilize the aggregate raw scores from each item.

Complex multiple-choice items

Complex multiple-choice items (Haladyna, 1997) are multiple-choice items wherein an initial set of possible answers to the item stem are presented, and then the actual response alternatives are defined as sets of these answers (where only one set contains all correct responses).

Example:

Which of the following descriptive statistics are measures of central tendency?

1 = Mean, 2 = Variance, 3 = Mode, 4 = Range

A). 1 and 3 B). 2 and 4 C). 1 only D). 1, 2 and 3

Although not identified in the literature, a possible variation on this format could utilize learner-generated answer sets as response alternatives. The learner would have the option of including or excluding each answer option individually, and the total number of response alternatives would be 2^n, where n is the original number of answers.

Example:

Which of the following descriptive statistics are measures of central tendency?

Mean Measure of central tendency // Not a measure of central tendency

Variance Measure of central tendency // Not a measure of central tendency

Mode Measure of central tendency // Not a measure of central tendency

Range Measure of central tendency // Not a measure of central tendency

True-false items

Osterlind (1998) estimates that the true-false item format is second only to the multiple-choice format in terms of frequency of usage on professionally-prepared tests. He identifies two sub-types of the true-false format: the right-wrong subtype, where the item stem is a declarative sentence; and the yes-no subtype, where the item stem is an interrogative sentence.

Examples:

The mean is a measure of central tendency. TRUE / FALSE

Is the variance a measure of central tendency? YES / NO

When separating the logic from the data of an assessment item for the purposes of computer-based delivery, however, semantic differences between content elements may or may not be relevant. For example, a data model for assessment content may distinguish between an item stem (e.g., "The mean is a measure of central tendency") and a set of response alternatives (e.g., "TRUE", "FALSE"), but not between the grammatical form of the item stems (i.e., between declarative and interrogative sentences). This illustrates an aspect of the decisions made in developing a data standard, and provides a possible entry-point for analytically evaluating them in terms of assessing the impact of distinguishing between certain types of content but not others.

True-False Items with Corrections

A variation of the true-false format identified (although not recommended) by Osterlind (1998) consists of a conventional true-false item with the addition of the directive that the learner should supply the correct answer if s/he believes the supplied statement is false.

Example:

The mean is a measure of central tendency.

TRUE / FALSE (The mean is a measure of ____________)

Completion Items

The category of true-false items with completions provides a transition between the classical selected-response item types such as true-false and multiple-choice and another category which can be labeled as 'completion' items. Per Osterland (1998), the category of completion items includes item types such as fill-in-the-blanks, sentence-completion, and short-answer. From the perspective of computer-based assessment with automated evaluation, however, the short answer item type is perhaps best excluded from this category. The rationale for this is that simple forms of completion items such as sentence completion and fill-in-the-blank generally have a limited set of possible articulations of acceptable answers, such as in the following example:

The mean is a measure of central ________.

For this item, the only acceptable answer is 'tendency'. Therefore, the evaluation of a response to this item could be automated without requiring some form of natural language processing.

CLOZE Procedure Items

The Wilson Taylor CLOZE procedure (Taylor, 1953) for evaluating the readability of a text has been applied to the development of assessment items for measuring text retention and comprehension (Osterland, 1998). In this item form, a text passage is presented with several words or phrases deleted. The deletions may occur at fixed intervals (e.g., every tenth word is deleted), or may be chosen based on the subject matter of the text (e.g., key domain terms are deleted). The respondent is directed to complete the sentences by entering the deleted words or phrases.

Example (source for original text: Abrami, Cholmsky & Gordon, 2001, p.381):

In the paragraph below, fill in each blank with the missing word.

In place of the independence assumption, the Oneway Within-Groups ANOVA relies on the _______ assumption (also known as the sphericity assumption). This assumption concerns the nature of the dependency among participants' scores across levels of the treatment (A) when there are ___ or more levels. Except for chance fluctuation, one would expect the correlation among participant scores for any two levels of the treatment to be _____.

Numerical Items

Numerical items are logically analogous to completion items, except that the response will be in numerical format.

Example:

The following data were gathered during an experiment:

Treatment group: 11, 9, 12, 9, -2, 7, 7, 16, 7, 10, 2, 13, 15

Control group: 9, 7, 7, 4, 8, 8, 7, 6, 3, 11, 6, 8, 5

What is the mean of the treatment group? _____

Short-Answer and Essay Items

In short-answer and essay items, the respondent is directed to write a text response to the item stem.

Example:

List and describe the threats to internal validity in a one-shot case study.

Although items in this category vary in terms of the specificity with which the boundaries of acceptable responses can be defined, as a whole they are currently beyond the bounds of automated evaluation through existing natural language processing technologies. A possible partial solution to this challenge may arise through the implementation of latent semantic analysis (Rehder, Schreiner, Wolfe, Laham, Landauer & Kintsch, 1998; Shapiro & McNamara, 2000), which would allow for the limited inclusion of these item forms in computer-administered environments where automated response evaluation is required.

Item Sets

Haladyna (1997) uses the term item set for assessment mechanisms involving multiple items associated with a common stimulus. Thus, the full context for each item is provided by that item's stem in conjunction with the common material shared by the items in that set.

Example:

The following data were gathered during an experiment:

Treatment group: 11, 9, 12, 9, -2, 7, 7, 16, 7, 10, 2, 13, 15

Control group: 9, 7, 7, 4, 8, 8, 7, 6, 3, 11, 6, 8, 5

1). What is the mean of the treatment group?

A). 8.92 B). 9.07

2). What is the mean of the control group?

A). 6.85 B). 7.42

A variation on the basic item set format is called the 'problem-solving item set' by Haladyna (1997) and the 'sequential' or 'step' model by Tutz (1997). In this format, items are sequenced in terms of their role in a problem-solving process. This provides a means for awarding credit for partial solutions of a given problem.

Example:

(Using the same stem and first two questions as the previous example)

3). What is the standard deviation of the treatment group?

A). 4.59 B). 4.97

4). What is the standard deviation of the control group?

A). 1.97 B). 2.12

5). What is the calculated z value for the mean difference between the two groups?

A). 0.38 B). 1.39

Multiple-Attempt, Single-Item Tests

Spray (1997) distinguishes between single-attempt multiple-item tests, such as the conventional multiple-choice test, with the multiple-attempt single-item test, where the learner has multiple tries to achieve a single goal and is evaluated on the number of successes. This is a common format in the evaluation of psychomotor skills, such as the number of free throws made in basketball. In computer-based assessment, this format may be useful in testing of hand-eye coordination tasks (e.g., simulations of aircraft landings).

Matching Items

Osterland (1998) describes the matching format as one where elements in a set of premises are associated with elements in a corresponding set of responses. This item form could also be labeled categorizing or classifying, depending on the context of the item stem.

Example:

Directions: Match the measures on the left with the properties on the right.

Measure	Attribute
Mean	Measure of central tendency
Mode	Outliers have a large effect
Median	Is the most-frequently occurring score
Range	

The basic components of a matching exercise are two sets of elements, with the respondent's task being to identify implied or explicit inter-set relationships. Different matching exercises may enforce one-to-one, many-to-one/one-to-many, or many-to-many links between sets. Additional variations can be generated by omitting set labels or not visually demarcating set membership, and by including elements in one set which do not have the prescribed relationship with the complementary set.

Example (one-to-one, set membership not identified):

Directions: Match the parametric tests with their nonparametric equivalents.

Correlated samples t test	Friedman ANOVA
Within-Groups ANOVA	Mann-Whitney U test
Wilcoxon signed-ranks t test	Oneway ANOVA
Kruskal-Wallis H test	Student's t test

Example (many-to-one, not all objects have links):

Directions: Match the measures on the left with the correct category on the right. Note: some of the measures may not belong in either category.

Measure	Category
Mean	Measure of central tendency
Variance	Measure of dispersion
Mode	
Range	
Effect size	

Ordering Items

In ordering-type items, a single set of objects is rank-ordered along an ordinal measurement scale. A common example of this type of item is the sequencing format, where the respondent is directed to arrange the objects into the correct temporal sequence. In print-based testing instruments, this type of item will typically have learners write in a number representing the rank beside each object.

Example:

Place the following steps in hypothesis testing into the correct order. Write 1 beside the first step, 2 beside the second step, and so on.

__ Compute the calculated value of the test statistic

__ State the null and alternative hypotheses

__ Find the critical value of the test statistic

__ Select the appropriate test statistic

__ Compare the calculated and critical values

__ Select alpha

__ Make a decision about the null and alternative hypotheses

In interactive media, the objects themselves could be rearranged. In this case, rank is represented spatially, either top-down (by convention, top = higher rank or earlier in temporal sequence) or left-right (left = higher rank or earlier in temporal sequence). These conventions may be culture-sensitive, of course. By separating the assessment logic and data from its presentation form, however, localization of an assessment instrument can be made more efficient. Thus, in an ordering item, the abstract 'order' is represented in the item's logic, whereas the specific spatial representation of the order could be left to the delivery system to implement on a context-specific basis. As distributed assessment systems are intended for open, potentially global audiences, the value of this flexibility is non-trivial.

Given the ability to manipulate media objects in computer-based assessment environments, a variant on the ordering-type item involving the use of an interval or ratio scale of measurement becomes possible. Consider the following two examples:

Example (ordinal):

Arrange the following inventions by date of discovery, so that the left-most item is earliest and the right-most item is the most recent:

Transistor Packet-switching network Semiconductor

Example (ratio):

Place the following inventions in their correct location on the timeline provided:

1920 1930 1940 1950 1960 1970 1980 1990 2000

Transistor

Packet-switching network

Semiconductor

The first example could be scored by simply evaluating the order of the object set (e.g., {1, 3, 2}), whereas the second would require a more complex scoring mechanism, since their order might be correct without the temporal location of the objects relative to the timeline being correct.

Configuration Items

A configuration is "an arrangement of parts" and a configuration task is "a problem-solving activity that selects and arranges combinations of parts to satisfy given specifications" (Stefik, 1995, p.608). Configuration items can be understood as a superclass of ordering items, where the operations the learner can perform and/or the evaluation function for the solution are potentially more complex than in unidimensional ranking. For example, in the traditional Magic Square problem, learners must use each of the numbers {1...(n*n)} to fill the cells of a (n x n) grid so that all rows, columns and both main diagonals sum to the same number.

Another variant on the configuration item places constraints on the operations the learner makes in obtaining the solution. This variant also differs from the previous form in that there are intermediary states or 'turns' that must be passed through in progressing from a given initial state to a goal state. The classic Cannibals & Missionaries puzzle is an example of this type of problem.

Example:

Three missionaries and three cannibals seek to cross a river using a boat. The boat can carry up to two people at a time. Both the missionaries and the cannibals are able to navigate the boat. The cannibals cannot outnumber the missionaries on either side of the bank at any time, because then the cannibals will eat the missionaries.

The logic for this type of problem could be represented through two sets, representing the two sides of the river bank. The initial state could be represented as {MMMCCC}, {}, and the goal state as {}, {MMMCCC}. Legal moves would be those that moved one or two elements from one set to the other, without leaving either set in the situation where the number of C elements was greater than the number of M elements.

Configuration items differ from simpler item types in that the evaluation algorithm may need to be located outside the assessment logic module; that is, it may be black-boxed from the perspective of the scoring algorithm inside the problem. This will become increasingly necessary as the algorithm becomes more complex, which will be the case for most problems that are not of the 'puzzle' variety such as those described above. For example, respondents might be directed to use a GUI electric circuit simulator to construct a simple circuit using a battery and two resistors, where the two resistors are in parallel with one another. When they have constructed their circuit, the circuit simulator could evaluate the circuit and pass messages (e.g., "parallel circuit", "series circuit", "broken circuit", "short circuit") to the scoring algorithm within the problem, which would process these messages in terms of the context of the given problem (e.g., only "parallel circuit" would be scored as correct). This would allow items of this form to be included in a data model, without requiring the data model itself to include sufficient functionality to implement complex simulations.

Branching Items

The increased complexity of moves evidenced in configuration items, as compared to matching or ordering items, can also permit a distinct item-type that could be labeled 'branching'. In a branching item, information necessary for deriving the final response to the item is obtained by progressive decision-making by the respondent. The information revealed at a given point in the item is dependent on the decisions made earlier. An example of the branching item might be a text-based representation of a diagnostic interview between a doctor and a patient, where the learner takes on the role of the doctor. On the first frame of the item, the patient might present with a brief narrative containing descriptions of several symptoms. The respondent would then have a choice of questions to ask, and based on the question selected, a second frame would be presented which would contain the answer to the selected question. This second frame could have a different set of possible follow-up questions to ask, which would determine the third frame selected. This process might continue for a set number of frames, or until the respondent felt ready to select a diagnosis for the patient's condition. Although logically similar in certain respects to the configuration problem-type, branching problems differ in that the learner does not have perfect information at the outset of the problem.

Simulation Items

The medical scenario described above is an example of a discrete branching item, as the frames would be discrete entities selected from a relatively small set of pre-authored frames. A more complex branching item would involve a system where the possible problem states would be much more numerous, so that the concept of frames as discrete entities would be supplanted by a more continuous conception. For example, if the learner was placed in the role of an air traffic controller, operations might include changing the speed and/or direction of any of the airplanes approaching the airport, and timing the take-offs of departing planes. The magnitude of the possible states in a problem of this nature (i.e., the possible locations and trajectories of the airplanes at any given point in time) prohibits the use of discrete representations of the problem space. Therefore, a more appropriate term for these types of items might be 'simulation', since they are generally associated with the operation or troubleshooting of a device or the management of a system.

Frequently, these simulations also operate on a continuous timeframe, unlike the medical scenario discussed previously where there is a 'turn-taking' interchange between the learner and the problem. In the air traffic controller simulation, for example, the simulation might proceed on an approximation of 'real-time', and the number of actions a learner might make would likely be far less than the total possible opportunities for action.

The set of item types described above provides a metric with which to judge the scope of the QTI data model. This analysis is performed following the presentation, below, of a brief overview of the data model.

The IMS Question & Test Interoperability Data Model

Introduction

The IMS Question & Test Interoperability Specification is intended to provide a data model for defining the structure and functionality of computer-based assessment instruments, from the macro level of an entire assessment down to the micro level response mechanisms that respondents use to enter responses to individual items. As can be gleaned from the title, the QTI constitutes an attempt to formalize and standardize the computer-based representation of assessment systems in order to increase their interoperability. In principle, the QT&I enables assessment systems that conform to the standard to seamlessly exchange information. This provides a protocol which would allow, for example, a learning management system to search a network for assessment items appropriate for undergraduate statistics students, retrieve these items, and integrate them into an automated assessment instrument.

The QTI data model is object-based, with the relationships between objects encoded within a XML (Extensible Markup Language) Document Type Definition (DTD). The logic and content of an assessment instrument is therefore expressed through a system of tags, similar in many ways to the specification of an online document though Hypertext Markup Language (HTML). To demonstrate how the QTI data model represents the structure and functionality of test items, two examples are provided below. Note that the examples below are script excerpts, not complete scripts. Certain elements that are essential for execution but are not relevant for demonstrating the assessment logic embodied in the script have been omitted for the sake of increased clarity. Consequently, the excerpts would not be correctly processed in a QTI-compliant XML reader as presented.

In the first example, the script excerpt demonstrates how the presentation aspects of a simple multiple-choice item with four response options would be represented through the QTI data model.

Example 1:

Presentation:

Which of the following is a measure of dispersion?

A). mean

B). standard deviation

C). mode

D). percentile rank

QTI:

```
<response_lid ident="MC1">
<material> Which of the following is a measure of dispersion?</material>
<render_choice>
<response_label ident="1"><material>mean</material></response_label>
<response_label ident="2"><material>standard deviation</material></response_label>
<response_label ident="3"><material>mode</material></response_label>
<response_label ident="4"><material>percentile rank</material></response_label>
</render_choice>
</response_lid>
```

In this basic example, only four types of tags are used: <response_lid>, <material>, <render_choice>, and <response_label>. The first tag, <response_lid>, designates the following content as corresponding to an item in the 'logical identifier' format ('lid' = logical identifier), which is the category generally intended for use with item formats such as true-false and multiple-choice. The <response_lid> tag has an 'ident' attribute contained within it, which ascribes a value to the item for identification purposes. In this example, the item is given an identification value of "MC1". This provides a mechanism for referring to this item from other methods. The next tag, <material>, indicates that the following text is content to be used in the presentation of the item. Since it comes immediately after the declaration of item format (i.e., <response_lid>), this content is to be processed as the item stem. The end of the content is indicated by the closing tag </material>. Therefore, in this example, "Which of the following is a measure of dispersion?" has been designated as the item stem.

The <render_choice> tag designates the beginning of the list of response alternatives for the item (its counterpart, the </render_choice> tag, designates the end of the list). Each separate response alternative is contained within the tag pair <response_label> and </response_label>. As with <response_lid>, the <response_label> tag also has an ident attribute associated with it, which ascribes a value to each response alternative for identification purposes. As with the item stem, the text content associated with each response alternative is enclosed within <material> and </material> tags. The final tag, </response_lid>, indicates the end of information related to the initial presentation of this item.

In the next example, the script excerpt demonstrates more complex structure and functionality. This example contains two items, one in multiple-choice format and one in numerical format (i.e., where the respondent supplies a numerical answer to the item). Response processing is also incorporated into the example, in that the respondent's answer to each item is compared to the correct answer. The result of this comparison determines the feedback message displayed to the respondent, as well as the adjustment to a scoring variable.

Example 2:

Presentation:

(In this example, respondent responses are represented with square brackets; e.g., [Learner enters 0.05])

Question 1:

For the next question, choose only one answer.

Which of the following is a measure of dispersion?

( ) mean

( ) standard deviation

( ) mode

( ) percentile rank

[Learner selects answer option 2, standard deviation]

Standard deviation is correct.

Question 2:

Express your answer to the following question as a decimal, not as a percentage (e.g., 0.5, not 50%)

What is the probability of a normally-distributed score occurring between the z-scores of −1.96 and +1.96?

[Learner enters 0.05]

0.05 is the probability of a score occurring *outside* the area bounded by –1.96 and +1.96.

Your answer is incorrect.

The correct answer is 0.95

QTI:

```
<questestinterop>
<section title="Fundamentals of univariate statistics #1" ident="FOUS1">
<item title="Identify measure of dispersion" ident="Dispersion" maxattempts="1">
<itemrubric view="Candidate"><material>For the next question, choose only one
answer.</material></itemrubric>
<presentation>
<response_lid ident="IMDA1">
<material> Which of the following is a measure of dispersion?</material>
<render_choice>
<response_label ident="mean"><material>mean</material></response_label>
<response_label ident="stddev"><material>standard deviation</material></response_label>
<response_label ident="mode"><material>mode</material></response_label>
<response_labelident="rank"><material>percentilerank</material></response_label>
</render_choice>
</response_lid>
</presentation>
<resprocessing>
<outcomes>
<decvar varname="WEIGHTED_SCORE" vartype="Integer" defaultval="0">
</outcomes>
<respcondition>
<conditionvar><varequal respident="IMDA1">stddev</varequal></conditionvar>
<setvar action="Add" varname=" WEIGHTED_SCORE ">1</setvar>
```

```
<displayfeedback feedbacktype="Response" linkrefid="I01_IFBK01">
</respcondition>
</resprocessing>
<itemfeedback title="Correctanswer" ident="I01_IFBK01">
<material>Standard deviation is correct.</material>
</itemfeedback>
</item>
<item title="Find percentage given area" ident="Percentage" maxattempts="1">
<itemrubric view="Candidate"><material>Express your answer to the following question as a decimal, not
as a percentage (e.g., 0.5, not 50%).</material></itemrubric>
<presentation>
<response_num ident=" FPGAA1">
<material> What is the probability of a normally-distributed score occurring between the z-scores of –1.96
and +1.96?</material>
<render_fib fibtype=”decimal” prompt=”box” maxchars=”4”>
<response_label ident="A"></response_label>
</render_fib></response_num></presentation>
<resprocessing>
<respcondition>
<conditionvar><varequal respident=" A">0.95</varequal></conditionvar>
<setvar action="Add" varname=" WEIGHTED_SCORE ">1</setvar>
<displayfeedback feedbacktype="Response" linkrefid="Correct"/>
</respcondition>
<respcondition continue=”yes”>
<conditionvar><varequal respident=" A">0.05</varequal></conditionvar>
<displayfeedback feedbacktype="Response" linkrefid="Explanation1">
</respcondition>
```

```
<respcondition>
<conditionvar><not><varequal respident=" A">0.95</varequal></not></conditionvar>
<displayfeedback feedbacktype="Response" linkrefid="Incorrect">
<displayfeedback feedbacktype="Response" linkrefid="ShowCorrectAnswer">
</respcondition>
</resprocessing>
<itemfeedback ident="Correct"><material>Your answer is correct.</material></itemfeedback>
<itemfeedback ident="Incorrect"><material>Your answer is incorrect.</material></itemfeedback>
<itemfeedback ident="ShowCorrectAnswer"><material>The correct answer is
0.95.</material></itemfeedback>
<itemfeedback ident="Explanation1"><material>0.05 is the probability of a score occurring *outside* the
area bounded by -1.96 and +1.96. </material></itemfeedback>
</item>
</section>
</questestinterop>
```

This example differs from the previous script excerpt in several notable ways. The entire script in the example is enclosed within <questestinterop></questestinterop> tags, which define the contained information as belonging to the QTI data model, in the same way that <HTML></HTML> tags are used in HTML to indicate that the information in a given file should be processed as HTML code. Another difference is that this example contains multiple items. A new structural level therefore becomes necessary to represent the instructional unit which is the union of the items within it. This level is the section, and its associated content is enclosed in <section></section> tags. Section tags also allow titles (e.g., “Fundamentals of univariate statistics #1”) and identification labels (e.g., “FOUS1”) to be associated with a section by setting the appropriate attribute values within these tags.

Item tags (<item></item>) are used to delineate individual items within the section. The entire set of information associated with any given item is therefore encased within separate <item></item> tags. In addition to titles and identification labels, Item tags can also contain a ‘maxattempts’ attribute, which specifies the maximum number of attempts the respondent can make in responding to the question. The additional information within the set bounded by the <item></item> tags is decomposed into four subsets:

- information related to the directions provided to the respondent for responding to the item, enclosed in <itemrubric></itemrubric> tags
- information related to the initial presentation of the item, enclosed in <presentation></presentation> tags
- information related to the processing of responses to the item, enclosed in <resprocessing></resprocessing> tags
- information related to feedback messages for different categories of responses, enclosed in <itemfeedback></itemfeedback> tags

Item presentation for the multiple-choice item in the second example is similar to that in the first example. The response processing section (encased within <resprocessing></resprocessing> tags) follows the presentation section. Response processing contains the information necessary to evaluate the respondent's responses. The main subunits within response processing are variable declarations, which are encased within <outcome></outcome> tags, and response conditions, which are encased within <respcondition></respcondition> tags. Variable declarations allow variable data, such as the number of items a respondent has answered correctly, to persist between items and sections. Response conditions associate system reactions with responses by evaluating responses to determine if they match given specifications.

In the first item of the provided example (i.e., the multiple choice question), a variable named "WEIGHTED_SCORE" is declared as an integer-type variable with a default value of 0. Then, a response condition is defined. The value of the respondent's response to the multiple-choice question (previously assigned the identification label "IMDA1") is checked to determine if it equals "stddev", which is the identification label of the correct response for this question. The <conditionvar></conditionvar> tags encase the expressions to be evaluated for this response condition, and the nature of the comparison operation is indicated by the <varequal></varequal> tags. In the case of <varequal>, the expression evaluates as TRUE if the respondent's response equals the value between the <varequal></varequal> tags.

If the expressions for the response condition are satisfied, the remainder of the information in the response condition is executed. For the first item, this corresponds to two actions. In the first action, the variable WEIGHTED_SCORE is incremented by 1. This is accomplished through a set variable command (represented by the <setvar> tag), with a specified action parameter value of "add". In the second action, a 'display feedback' process is triggered using the <displayfeedback> tag. The substance of the feedback to display to the respondent is referenced by the linkrefid parameter value (in the example, the value is "I01_IFBK01"). The feedback content itself, however, is not contained within the response condition, but is defined after the response processing data. Each distinct feedback message is enclosed within separate <itemfeedback></itemfeedback> tags. Within these tags, an identification value (e.g., "I01_IFBK01") and content (e.g., the text string "Standard deviation is correct.") are associated to one another. Then, if the event "displayfeedback" with a parameter value of "I01_IFBK01" is triggered within response processing for this item, the system can identify the content associated with this event.

The second item in the second example demonstrates the functionality for presenting and processing an item wherein the respondent enters a numerical response. The item is first designated as being in numerical format through the <response_num> tag (as opposed to the <response_lid> tag used in the preceding multiple-choice item). The mechanism through which the respondent enters a response is defined by the <render> tag-type, which in this case is a <render_fib> tag, where 'fib' stands for 'fill in the blank'. The attributes in the render tag specify that the respondent will be presented with a box wherein numbers and a decimal point can be entered, and that the maximum number of characters in this box is four. The response the respondent enters will be referenced through the identification value of "A".

In the response processing component for this item, the response is compared to the correct answer, which in the case of the example supplied is 0.95. If so, the WEIGHTED_SCORE variable is incremented by one, the feedback message labeled "Correct" is presented, and the item concludes. If not, the answer is compared to 0.05, which corresponds to the respondent entering the area outside the Z scores of +/- 1.96. In this event, a customized feedback message is presented: "0.05 is the probability of a score occurring *outside* the area bounded by –1.96 and +1.96." Regardless of whether this response condition is evaluated as true or not, the final response condition is still evaluated, as the 'continue' attribute of the middle response condition was set to "yes". The 'continue' attribute specifies whether subsequent response conditions should be evaluated if the current response condition is true. In the final response condition, any answer other than 0.95 (i.e., the correct answer) triggers the presentation of a feedback message indicating the answer was incorrect, as well as the correct answer itself.

Overview of the QTI Data Model

The above script excerpts provide an example of the manner in which the QTI specification instantiates the logic and data associated with simple computer-based assessment items. Broadly speaking, the data model is responsible for specifying the following functionality:

- Presentation of the item's directions and stem to the respondent, along with a mechanism for entering a response to the item (e.g., display of a multiple-choice item with four response alternatives, along with a means for selecting one of the answers).
- Categorization of the response in terms of possible predefined events (e.g., correct answer, incorrect answer).
- Triggering of system and respondent feedback based on the response event(s) that occurred (e.g., if the answer was correct, add one to the total score for the evaluation).

These categories of functionality are now discussed in turn.

Item Presentation and Response Mechanism

Response-types are the templates for structuring the information necessary to elicit and obtain a response from a respondent. For example, in a conventional multiple-choice item, the response-type carries the information necessary to define the item as a multiple-choice type question, to list the response alternatives presented to the learner, to specify whether the respondent is allowed to select single or multiple responses, and so forth. All response-types share the following common information elements:

- An identification attribute, allowing other parts of the item (such as response processing) to identify the response data from each response-type.
- A cardinality attribute, which specifies the number of members in the response data set, and whether the order of the members in the set is significant.

- A timing attribute, which specifies whether the duration between item presentation and a respondent's response is to be recorded.
- A render-type, which specifies the nature of the mechanism through which responses to the item are entered.

In addition to these common elements, the different response-types also contain specific information necessary for different forms of response data. In the current version of the specification, five response-types are provided: string, numeric, x-y coordinate, logical identifier, and logical group.

String, numeric, and x-y coordinate response-types return data that corresponds literally to the content of the underlying response. For example, for a simple arithmetic question asking "What is 2 + 2?", a response of {4} would indicate that the respondent's response to the question was the numeral 4. The string response-type handles text-based (alphanumeric) responses, whereas the numerical response-type handles numeric responses in integer, decimal and scientific notation formats. The coordinate category handles responses wherein the respondent points at a location onscreen, which is encoded as a point through X and Y coordinates.

The two 'logical' response-types, logical identifier and logical group, utilize response mechanisms wherein identification labels are assigned to possible responses, and response data is received as a set of one or more of these identification labels. For example, in the following multiple-choice question, identification labels would be assigned to each of the possible responses:

Which of the following is a measure of dispersion?

mean median mode variance

If the identification labels consisted of 'Answer1' for mean, 'Answer2' for median, 'Answer3' for mode, and 'Answer4' for variance, then the correct response data set for this item would be {Answer4}. Thus, in a 'logical' response-type, the answer is processed as an identification label (e.g., {Answer 4}) rather than as semantic content (e.g., {"variance"}), as was the case with the three previous response-types.

The cardinality attribute allows for the possibility of responses that consist of multiple object labels. For example, in a multiple-choice question where the respondent was directed to select all the answer options that were correct, the cardinality attribute would be set to 'multiple' to allow for a response set with more than one logical identifier in it. If the order of the logical identifiers is important as well, then the cardinality can be set to 'ordered'. This allows, for example, for questions that direct the respondent to order a set of answer options in terms of their causal sequence. In all of these cases, the response set is a single set of individual data elements, such as {Answer2, Answer3, Answer1, Answer4}.

For item forms whose response data consists of multiple elements in more complex relationships than a single ordered set, the QTI specification provides the logical group response-type. Although this is the most incompletely defined response-type in the current version of the specification, it presumably adds the ability to receive multiple sets of data elements. This would be useful for matching questions, such as: Match the parametric tests with their nonparametric equivalents.

(A) Correlated samples t test	(1) Friedman ANOVA
(B) Student's t test	(2) Kruskal-Wallis H test
(C) Oneway ANOVA	(3) Wilcoxon signed-ranks t test
(D) Within-Groups ANOVA	(4) Mann-Whitney U test

A possible response data set for this question would be {{A,1}, {B,4}, {C,2}, {D,3}}, with each inner set representing a single matched pairing.

In order to receive response data from a respondent in a computer-based environment, a means for entering data must be defined. This is accomplished in the QTI data model through the selection of a render-type. The basic form of response mechanism rendering is the 'choice'. This render-type is used for specifying the set of media objects that corresponds to the individual response alternatives available to the respondent. For example, a list of terms corresponding to the response alternatives for a multiple-choice question would be contained within a 'choice' render-type. The presentation system would handle the display of the list of terms, along with the provision of a mechanism for selecting one or more of the terms (e.g., checkboxes beside each term that the respondent would click on to select that term as a response). The choice render-type also allows the item author to specify whether the order in which the media objects are presented should be randomized or not.

In addition to the 'choice', the current version of the QTI data model supports three additional render-types: hotspot, fill-in-the-blank, and slider. The hotspot render-type enables response mechanisms involving the selection of predefined regions in the onscreen display. For example, if an image of a bar graph was displayed onscreen along with directions stating that the respondent should click on the bar corresponding to the data set's mode, hotspots could be defined on each bar so that the item could encode mouse clicks made by the respondent on the diagram. Hotspots are not restricted to static graphics, however, as the QTI data model allows this render-type to be associated with text and video as well.

The fill-in-the-blank (FIB) render-type is associated with string and numerical response-types where the respondent enters a response through some form of alphanumeric input device, such as a keyboard. The FIB object specifies the type and length of data allowed as a response, the manner of its encoding, and the visual presentation of the onscreen data entry widget (e.g., a box). Multiple FIBs can be integrated into a single response-type, potentially allowing for items with multiple or ordered cardinality.

As an alternative to FIBs, the QTI data model provides a useful response mechanism in the form of the slider. Sliders are common GUI input widgets which allow computer users to select numerical values from within a continuous range by moving a pointer along a scale (Fowler, 1998). An advantage of using a numerical slider over numerical entry (e.g., the respondent entering digits on a keypad) is that the slider can be used to restrict inputs to certain kinds of values (e.g., integers between 1 and 100). This can eliminate the necessity of checking the response for certain types of errors that may be unrelated to the respondent's ability to correctly answer the item, such as the use of decimals when an integer response is indicated, rounding errors, and so forth. The QTI data model allows a slider response mechanism to be associated with both numerical and logical identifier response types. For both response types, the slider's scale is defined by specifying its lower and upper bounds, as well as the granularity of the steps between these endpoints. The presentation of the slider can also be specified in terms of its orientation (i.e., horizontal or vertical), whether its steps are labeled, and the starting position of the slider. Sliders can be configured for single or multiple responses. If the slider is defined as part of a numerical response type, the response data for the item will be received as numerical data. If the slider is defined as a logical identifier response type, then the response data will consist of a logical identifier corresponding to the region the slider pointer is in. These regions can be defined as exact values (e.g., 1.42) or ranges of values (e.g., between 1.4 and 1.5).

Response Processing

Once a response data set has been received for the item, it must be processed by the assessment system. At the Item level, response processing consists of grouping possible respondent responses into one or more categories, and associating these categories with two kinds of operations: changes in variable values, and the presentation of feedback to the respondent, such as hints and solutions. For example, in a multiple-choice question with four response alternatives where only one is correct, two response categories might be defined. The first would contain only the correct answer, whereas the second would contain the three incorrect answers. If the respondent answered the question correctly, his or her response would therefore fall into the first category. Any events that were linked to this category would then be triggered. This might be an operation on a variable (e.g., incrementing the value of the CORRECT_RESPONSE variable by one), and/or the display of feedback to the respondent (e.g., "Correct!"). Conversely, if the respondent answered the question incorrectly, events associated with the second category would be triggered (e.g., displaying the message "Incorrect!" to the respondent). More complex feedback schemes can be implemented through the definition of additional response categories.

Before responses can be processed, however, a mechanism for storing information across items must be specified. The first aspect of response processing therefore involves defining persistent variables, so that information about individual item responses can be accumulated during an assessment session. The variable types supported by the QTI are integer, decimal, scientific, string, Boolean, enumerated, and set. The first five types correspond to the conventional variable types available in many programming languages. The enumerated variable type allows for a set of possible values to be defined when the variable is declared (e.g., {red | yellow | green}), with the variable allowed to take on any single value of this set at a given time. The set variable type allows a variable to be treated as an open set of elements. Although relatively undocumented, presumably this variable type provides a means for incorporating list and array functionality in future versions of the standard.

Once variables have been defined, each response category considered salient in terms of the evaluation is given its own response condition. As discussed previously, this may be as simple as a dichotomous separation of responses into correct and incorrect answers, or may involve more complex categorization schemes with multiple and overlapping levels. Regardless of the complexity of the overall processing scheme, the two fundamental components of a response condition are the specifications for determining whether a given response falls into that response condition, and specifications for what is to occur should the requirements for that condition be met.

The Condition class is the collection of methods for defining outcome events based on the response to an item. Using this class of methods, an item author can specify one or more comparison tests for each response category that will determine whether a given response is a member of that category. The most basic form of comparison test is one that compares a response to a single value, returning TRUE if the response is equal to this value. For text and numeric data, the response can also be evaluated with respect to whether it is a subset of the specified comparator value. Standard mathematical comparators such as <, <=, >, and >= are also provided for evaluating numerical data. For responses that are defined as a set of coordinate values (e.g., an onscreen point encoded as a pair of X and Y values), the varinside method provides a means for specifying a geometrical shape and testing whether the response coordinates lie within that geometrical shape.

In addition to providing methods for specific response modalites such as coordinate data, the Condition class also includes methods for handling generic response categories (e.g., unanswered items), meta-information about the response (e.g., the time taken to answer the question) and logical operators which allow multiple comparison tests to be combined into a single response condition. For generic response categories, two methods are provided: 'unanswered' and 'other'. If no response is entered by the respondent for a given item, the 'unanswered' method in a comparison test will return a value of TRUE. 'Other' provides a means for specifying a response condition that is TRUE only if all other conditions specified for the item are false. For timed items, condition tests can evaluate whether a response occurred within a specified time period. Operators available for this form of comparison are the same as those for numerical data. The standard also supports the logical operations AND, OR, and NOT. This provides a mechanism for linking together multiple tests within a single response condition. For example, the use of the AND operator allows for the specification of a variable within a range, i.e. (X > 17.0) AND (X <= 18.0).

Through the methods described above, the Condition class enables a wide variety of comparisons to be performed on response data that consists of a single data element. As stated previously, the QTI data model also provides a mechanism for processing answers consisting of multiple data elements. This form of processing is implemented through the cardinality attribute in the Response class. The cardinality attribute is an enumerated variable with possible values {single, multiple, ordered}. Items with a cardinality attribute of 'single' correspond to the simple case where the respondent's response to an item consists of only one data element, i.e. when the cardinality of the response data set is one.

If the cardinality of the response set is non-single, then a means for specifying the process for applying comparison tests to the response set is required. For example, consider a multiple-choice question where answer options A and C were correct, and both needed to be specified by the respondent in order for the response to be evaluated as correct. Within a single response processing condition, two tests would be defined, one evaluating the response set to determine if it contained A and the other to determine if it contained C. To indicate that each test is to be applied to the entire response set, the cardinality attribute is set to 'multiple'. Through the use of the other logical operators, such as NOT and OR, more complex response conditions can be defined, such as evaluating whether a response set contains A and C as well as any of B and E (e.g., (A AND C) AND (B OR E)).

The 'ordered' value for the cardinality attribute allows for the evaluation of response sets where order has meaning. For example, if a question asked for the steps of hypothesis testing to be placed in their correct order, the response set might be {3, 5, 2, 4, 1, 6, 7}. The tests listed within a single response condition are then evaluated against the individual members of the response data set, in the order in which they are presented. That is, the first test is applied to the first member of the response set, the second test is applied to the second member, and so forth.

If the comparison tests in a response condition have been satisfied, the actions associated with this condition are triggered. Two types of actions are provided in the QTI data model. The first action involves changes to variable values. Variable values can be adjusted through the Setvar class, which allows the specification of an operation (set, add, subtract, multiply, divide), a variable the operation is to performed upon, and a value to be used in the operation. This allows variables for tracking of such assessment session parameters as the number of correct responses made, the weighted examination score, and the number of incorrect attempts.

The second action that can be associated with the fulfillment of a response condition is the triggering of feedback events. One or more identification labels for feedback events can be listed within a response condition, and if the requirements for that response condition are satisfied, then each listed feedback event will be triggered. The actual substance of the feedback event is defined outside of response processing, however (discussed below).

If there is more than one response condition in a given item, flow control is determined by the 'continue' attribute. The default execution process of the QTI model is to individually process the response conditions in the order they are listed in, and terminate the process at the first response condition that returns a value of TRUE. Composite feedback may be constructed by specifying 'continue = yes' in one or more response conditions, which indicates that the process should not terminate if TRUE is returned for these conditions.

Defining Feedback Events

Essentially, feedback events are predefined associations between one or more media objects (e.g., a string of text) and an identification label. In the current version of the QTI data model, only feedback events within Items have been specified, but placeholder tags have been included for feedback events at the Assessment and Section levels (<assessfeedback> and <sectionfeedback>, respectively). These placeholder tags have been earmarked for elaboration in the next version of the standard.

The main components of a feedback event are:

- Identification label
- Content (as media objects)
- Type and style

As previously demonstrated in the script excerpts, the primary role of feedback events is to associate a set of media objects with an identification label. This identification label can then be referred to in the response processing data for that item. Feedback events are further defined by type and style tags. The possible values for feedback type are {response, solution, hint}. A 'Response' feedback type indicates that the feedback event is a message that comments on the nature of the respondent's response to the item, such as its correctness. The solution feedback type indicates that the content for the message is a solution for the item. Finally, the hint feedback type provides a mechanism for providing support to the respondent if multiple attempts are allowed for the item and the respondent's previous response(s) was/were incorrect.

For solution and hint type feedback, the style tag can be used to further specify the presentation of the feedback event. The possible values for style are {complete, incremental, multilevel}. Complete is the default value for style, and indicates that the content defined in the feedback event is to be considered as a unitary whole. Incremental is used for feedback content that is divided into parts and is intended for presentation in successive steps, such as an increasingly detailed sequence of hints. The sequence for presentation is defined by the sequence in which the content is listed in the subsequent content tags. Multilevel style is used for situations where multiple content units exist within a single feedback event, and each content unit is meant to be treated as a distinct entity. This provides a mechanism for providing an unordered set of hints, which can be individually selected for presentation on a random basis.

Macro Structure and Functionality of the QTI Data Model

The QTI data model utilizes a three-tiered hierarchy of Assessment, Section and Item (ASI) units to represent the macrostructure of an assessment instrument. As described above, the Item unit is used to refer to individual questions in an assessment instrument, whereas the Assessment unit is utilized as the top-level unit, and corresponds to an entire assessment. The Section unit is used to refer to levels between the Item and Assessment. In addition to containing Items, Sections can also contain one or more Sections. This allows the number of levels in the middle tier to be expanded as necessary. In principle, any ASI unit can be imported or exported across a distributed network by a QTI-compliant system, allowing both the retrieval of self-contained assessment instruments as well as the retrieval of individual Items and/or Sections followed by the automated assembly of these elements into a complete assessment instrument.

In the current version of the QTI data model, macro-level functionality is undefined, save for the existence of placeholder objects which indicate the general direction subsequent elaboration of the data model is expected to progress in. For example, both Items and Sections will have pre- and post-conditions associated with them, allowing the assessment system to branch through a collection of Items / Sections based on the responses to previous Items / Sections. Aggregated response processing across multiple Items or Sections has also been earmarked for implementation in the next version of the data model.

Assessment of the QTI Data Model

As stated previously, the evaluation of the QTI specification is conducted on two fronts: a comparison of the scope of the current version of the data model against the item types commonly used in print-based and computer-based assessment, and an investigation of the internal logic of the data model in terms of its structure and functionality. These are now discussed in turn.

Assessment of the Data Model's Scope

In its current version, the QTI data model can only be evaluated at the Item level. Specification of the logic and content at the Assessment and Section level is largely incomplete. Therefore, item types that are partly defined at the macro level, such as multiple-attempt and branching items, cannot be included in the current scope of the QTI data model. This does not preclude evaluation of the data model, nor its potential adoption by a given organization, as a viable Item-level specification is arguably a valuable asset in its own right. Therefore, the focus of the evaluation in this thesis is the implementation of individual item types in the data model. It bears mentioning that, at least in the case of multiple-attempt and branching items specifically, the additional functionality required for their implementation within the data model should be relatively trivial.

With the above caveat, the QTI data model is capable of implementing the majority of common assessment item types identified in the literature survey. It is particularly strong in terms of its ability to handle those staples of academic assessments, the true-false and multiple-choice item types. For example, the data model provides a mechanism for specifying whether the item's list of response alternatives should be subjected to randomized shuffling at delivery-time. This allows item authors to specify shuffling for multiple-choice item types (Appendix, example 1), while preserving the prearranged order of response alternatives in a rating scale or true-false item as necessary (Appendix, example 2). Another useful feature of the data model is that aspects that are constant across a given set of items can be referenced through URI pointers. Therefore, a rating scale (e.g., 1 = below average, 2 = average, 3 = above average) can be authored once and then reused in any given item by including a pointer to the location of the rating scale content (Appendix, example 3). This feature in and of itself could significantly increase the efficiency of a large-scale assessment content development and production process.

The QTI data model is also capable of implementing variations on the basic multiple-choice item type. Complex multiple-choice items are supported (Appendix, example 4), since they can be implemented in a logically identical manner to conventional multiple-choice items. Scoring models that give partial credit for responses that are partially correct can also be used (Appendix, example 4). Although there is no provision for randomized generation of distractor sets in these items, it is possible to create a set of alternatives with a dichotomous response associated with each response, which allows for respondent-generated response sets (Appendix, example 5). One problem that affects complex item types is that there is no explicit means to specify presentation requirements, such as line breaks between components of the item stem. Although this is understandable in a standard whose core principle is the separation of logic and data from presentation, there are aspects of presentation that are salient from an instructional design perspective. Some of these aspects may need to be categorized as attributes of an item type's logical structure rather than its presentation. This issue is further discussed in the assessment of the data model's structure and functionality that follows this section of the thesis.

The QTI data model provides adequate support for completion items that are amenable to simple forms of automated response processing, such as fill-in-the-blank (Appendix, example 6) and CLOZE-procedure item types (Appendix, example 7). A limitation on its support for CLOZE-procedure items is that it does not allow for automated fixed-interval deletion schemes, as all deletions must be prescribed at author-time. This is not a serious limitation, as fixed-interval deletion schemes are likely to be more common in readability evaluation than in educational assessment. Short-answer and essay items can be included in QTI-based assessments (Appendix, example 8), but naturally cannot be scored through the simple functionality provided by the QTI data model's equivalent of chained IF-THEN statements. One means for bypassing this limitation would be to call an external application for response processing. For example, the assessment engine could pass a short-answer response to a separate text analyzer module (perhaps employing Latent Semantic Analysis) and then receive a coded variable signifying different types of response events (e.g., a grade rating from A to F for the short-answer response). The QTI data model provides a mechanism for extending response processing in this manner.

For numerical problems, the QTI data model includes the functionality necessary to specify the appropriate response types (integer, decimal, or scientific), although the manner in which invalid entries by the respondent should be handled is not specified. This is arguably an aspect of an item's logic, and therefore should be included in the data model. Response processing for numerical items allows for the specification of response events in terms of intervals (Appendix, example 9) as well as in terms of points (Appendix, example 10).

Item sets can be created by incorporating multiple response-type definitions within a single item, and including common material in the item stem (Appendix, example 11). This also allows for the implementation of hybrid item forms such as true-false with corrections by combining the logical identifier response type with the fill-in-the-blank response type (Appendix, example 12). With lengthy item sets, the presentation system will need to provide a mechanism in the GUI for respondents to quickly access the common material in the item set. The ability to modify responses to previously-answered questions in a given item set may need to be restricted, however, in the case of computer adaptive testing where items within an item set are modified based on earlier responses to other items in the same item set (Green, Bock, Humphreys, Linn & Reckase, 1984).

Problem-solving item sets can be developed in a similar fashion to basic item sets, and response processing can compare the responses within the same item to each other (Appendix, example 13). There is no provision for using response data within formulas in response conditions, however, so noticeable constraints exist here. This limitation could be overcome to a certain extent in the case of automated item generation. An automated item generator could create item sets and enter the values into the corresponding slots in the item template. Example 13 in the Appendix demonstrates how slots could be defined through the labeling of response alternatives. This would allow the responsibility of implementing a diagnostic framework to be offloaded from the QTI data model to the automated item generator.

The limitations to the current QTI data model's scope begin to arise more severely in the case of more logically complex item types such as matching and ordering. Both of these item types are problematic when implemented in terms of moveable objects and logical groups. These limitations, and possible remedies, are further discussed in the assessment of the data model's structure and functionality that follows this section of the thesis. In the absence of these remedies, matching items can be implemented (Appendix, example 14) in a similar manner to complex multiple-choice items with learner-defined sets of response alternatives. Ordering items can be partially implemented using numerical entries (Appendix, example 15), but this precludes the ability to compare the relative ordering of the response data set (e.g., was step 4 placed before step 3). It also precludes the implementation of randomized shuffling of elements in the item.

It would be unreasonable to expect the QTI data model to provide a language for the specification of complex configuration and simulation items, as this would require it to incorporate a wide range of functionality, to the point where it could be classified as a programming language. This limitation can be partially bypassed by calling an external simulation engine, and receiving events from the simulation as response data (Appendix, example 16). Simulations that involve autonomous events in a continuous process, such as the continual motion of airplanes in an air-traffic control simulation, are not possible in the current version of the data standard.

Assessment of the Data Model's Structure and Functionality

The following sections focus on specific issues related to the data model's structure and functionality, and propose modifications to the standard that would remedy these issues where possible.

Response Process

As described previously, the basic response process at the Item level consists of four phases:

1. The assessment system displays the item, including all associated media objects and the response mechanism(s)
2. Using the response mechanism(s), the respondent enters and submits a response to the item
3. The assessment system evaluates the response and triggers the appropriate events (e.g., displaying a feedback message to the respondent)
4. The assessment system selects another Item and returns to step 1, or the assessment terminates

Although it is not specified in the QTI data model, the delivery system will need to be responsible for adding the necessary functionality for the candidate to pass control back to the assessment system when the process transitions from step 2 to step 3. For example, the delivery system may add a Submit Answer button to the generic assessment interface. After a candidate has completed entering the response for a given item, he or she would click on this button in order to signify that the response is ready to be processed. An undesirable alternative in the case of items with single cardinality response sets would be to have the delivery system automatically send the response to the assessment system as soon as the respondent clicked on an answer option. For example, in a multiple-choice question where only one answer option could be selected, a response could be submitted as soon as the respondent clicked on any of the answers. This would open up the possibility of user errors in mouse clicks leading to mistaken answers being recorded. In any event, a submission button would be necessary for any items with non-single cardinality where the number of elements in the response set was variable, as this context would preclude automated response submission on the basis of mouse clicks. Given that the usability of an interface is generally increased when users are allowed to explore its features in a non-punitive manner (Heckel, 1991), it is advisable that a Submit Answer button (or a widget with similar functionality) be added to all item types during delivery. This would be especially critical in the case of high-stakes assessment, where the consequences of usability-related design flaws are more serious.

A related issue in the control of the response process arises in the context of timed responses. Currently, the QTI data model provides an attribute ('rtiming') that allows an item author to specify whether the duration between item presentation and response submission is to be recorded. This duration can subsequently be included in response processing by creating conditions that combine a verification of a correct response with a check to determine whether the response occurred in a given time frame:

```
<respcondition><conditionvar><and>
<varequal respident="Answer">A</varequal><durlt respident="Answer">00:01:00</durlt>
</and></conditionvar></respcondition>
```

As the QTI data model follows the ISO8601 standard for the representation of dates and times, the above response condition evaluates whether the respondent's response was both "A" and occurred in less than ('durlt' = duration less than) one minute. A notable omission in the data model is the absence of an attribute value that specifies whether the response duration is displayed to the respondent in real-time. In some speed tests, for example, it will be desirable to continuously display the response time to the respondent. In other cases, such as when a weighted score model based on response duration is being applied without the respondent's knowledge, response time will not be intended for display. Therefore, a recommended revision to the 'rtiming' attribute is to that its range of values be changed from {yes, no} to {display, background, no}.

For the further specification of response timing, the QTI data model provides two additional data elements that can be associated with any ASI unit: duration and qmd_timelimit. The role of 'duration' is somewhat uncertain in the current version of the specification, as it is defined as both the duration "permitted" for the ASI unit to which it is attached, as well as the "expected duration" of that unit. In the first sense, there is an implication that this data element, if associated with an Item, would force a transition from step 2 to step 3 in the response process for that Item if its specified duration was exceeded. For example, the respondent might have 30 seconds within which to answer the Item or else the assessment system would record an 'unanswered' response. The ability to associate duration attributes with multiple levels of the ASI hierarchy is also useful, as this allows individual Sections or Items to be administered under set time limits. This functionality is critical for the scope of the QTI data model, as it enables the delivery of pure speed and hybrid speed/power tests, the later of which accounts for the majority of formal assessment contexts (Roskam, 1997). As was the case with the rtiming attribute, however, there is no current provision for specifying whether the duration should be displayed to the respondent or not. In most cases, it is probable that the respondent should be aware of the elapsed time, since the consequence of exceeding it would be the termination of the ASI unit. Therefore, at minimum, the QTI data model should specify that the delivery system has a responsibility for displaying the elapsed time to the respondent whenever a 'duration' is associated with an ASI unit. Ideally, the duration object would have an attribute added to it that controlled whether the elapsed time was displayed to the respondent.

The above recommendation presumes, of course, that the definition of the 'duration' object is in terms of the time permitted for that associated ASI unit, rather than expected duration. This assumption is supported when the time-related metadata elements for Items (qmd_timedependence) and ASI units (qmd_timelimit) are examined. Qmd_timedependence is a Boolean metadata bit (TRUE/FALSE) which describes whether responses are time dependent. This is likely meant to be associated with Items that use response duration in the processing of responses, that is, Items which have rtiming set to TRUE. For describing ASI units that have time limits through the 'duration' object, the QTI data model provides the Qmd_timelimit descriptor, which can take on either an integer value (i.e., the time limit in minutes) or the string "unlimited". Since the 'duration' object supports the ISO8601 format, there is an obvious type mismatch here with the qmd_timelimit element, which could prove problematic and should be rectified. A more serious shortcoming here is that there is no metadata descriptor provided for expected duration of individual ASI units. Given that one of the primary goals of the QTI data model is to enable efficient search and retrieval of ASI units for configuration into new assessment instruments, information regarding expected duration is valuable in that it allows for the estimation and control of assessment duration during the configuration process. It also allows for differentiating between multiple ASI units whose profiles are otherwise similar. For example, in the context of a computer-adaptive test operating with a distributed item bank, the item selection algorithm could weigh the expected duration of items against their information functions in order to determine the optimal item to use in terms of efficiency; that is, the item that generates the most information in the least amount of time. Since the mean response time to an item may be less than its time limit, using qmd_timelimit as the metadata element for this purpose would not accurate. Furthermore, the dual purposing of this element would likely engender confusion. Instead, it is recommended that the data model be extended to include a metadata descriptor labeled qmd_meanduration that could be associated with any ASI unit.

Items Permitting Multiple Attempts

In low-stakes computer-based assessment such as those associated with tutorials, it is relatively common for respondents to have more than one attempt at getting the correct answer for an Item. Often, a hint is provided after incorrect responses to guide the respondent towards the correct response. In the QTI data model, the maxattempts attribute in the Item object determines whether multiple attempts are allowable for the current item, and the number of attempts permitted. The action to be taken by the assessment system when the number of attempts exceeds this value is not specified in the current version of the QTI data model, however, and neither is the exact nature of response process control between attempts for multiple-attempt items. For example, consider a multiple-choice item with maxattempts =2, that is, one that allowed for two attempts at the correct answer. After the first incorrect attempt, a hint should be provided, and after the second incorrect attempt, the correct answer should be provided. In the case of correct answers, the item should present the message "Correct!" and then progress to the next item. Per the QTI data model, the following script should suffice to encode the desired functionality:

```
<item title="Identify measure of dispersion" ident="Dispersion" maxattempts="2">
<itemrubric view="Candidate"><material>For the next question, choose only one answer. You have two attempts to get the right answer.</material></itemrubric>
<presentation><response_lid ident="IMDA1">
<material> Which of the following is a measure of dispersion?</material>
<render_choice>
<response_label ident="mean"><material>mean</material></response_label>
<response_label ident="stddev"><material>standard deviation</material></response_label>
<response_label ident="mode"><material>mode</material></response_label>
<response_labelident="rank"><material>percentile rank</material></response_label>
</render_choice></response_lid>
</presentation>
<resprocessing>
<respcondition><conditionvar><varequal respident="IMDA1">stddev</varequal></conditionvar>
<displayfeedback feedbacktype="Response" linkrefid="CorrectAnswer"></respcondition>
<respcondition><conditionvar><other respident="IMDA1”/></conditionvar>
<displayfeedback feedbacktype="Response" linkrefid="IncorrectAnswer"></respcondition>
</resprocessing>
<itemfeedback ident="CorrectAnswer"><material>Correct!</material></itemfeedback>
<itemfeedback ident="IncorrectAnswer">
```

<hint feedbackstyle="incremental"><hintmaterial><material>Two of the incorrect answers are measures of central tendency, not dispersion. Try again.</material></hintmaterial></hint>

<solution feedbackstyle="incremental"><solutionmaterial><material>The correct answer is standard deviation.</material></solutionmaterial></solution>

</itemfeedback>

</item>

The main problem with the current version of the QTI data model is that there is no means of specifying the conditions under which the response process should be terminated in advance of the number of cycles defined by maxattempts. For example, in the context of the example described above, the response process should display "Correct!" and then terminate if the respondent obtains the correct answer on the first attempt. Although the identification label associated with the feedback for this response condition is "CorrectAnswer", this label has no semantic significance to the assessment system, as it is merely a pointer to the feedback content associated with this response condition. Therefore, in executing the above item, the assessment system would provide the respondent with a second attempt at the answer, despite the fact that he or she had obtained the correct answer on the first attempt. To add the necessary functionality, a new method needs to be added to the response condition class that would allow the item author to designate that response condition as a terminal condition.

A second problem in the processing of multiple-attempt items is that there is no means for incorporating the number of attempts into variable operations. This precludes giving respondents a higher score for an item if they obtain the correct answer on the first attempt instead of the second. A new set of methods needs to be added to the conditionvar class that would allow for the evaluation of the number of attempts as part of the response condition. This could be accomplished through methods that are similar to those provided for evaluating the duration of a response (i.e., durlte, durlt, durequal, durgt, durgte). This would allow items to trigger different events based on the number of attempts the respondent has made:

<respcondition><and>

<conditionvar><varequal respident="IMDA1">stddev</varequal></conditionvar>

<conditionvar><attlt respident="IMDA1">2<attlt></conditionvar></and>

```
<setvar action="add" varname="score">3</setvar>
</respcondition>
<respcondition>
<and><conditionvar><varequal respident="IMDA1">stddev</varequal></conditionvar>
<conditionvar><attlt respident="IMDA1">3<attlt></conditionvar></and>
<setvar action="add" varname="score">1</setvar>
</respcondition>
```

This script excerpt uses an 'attlt' method to determine whether the current attempt number was less than number specified. The respondent would receive 3 points for a correct answer on the first attempt, and 1 point for a correct answer on the second attempt. Another advantage of incorporating these methods into the QTI data model would be that feedback events could also be differentiated on the basis of which response attempt was currently being processed.

Hotspots

The hotspot render-type will likely prove to be a useful part of the QTI data model, as it allows static graphics and other media content to be displayed as integral wholes while simultaneously being subdivided into smaller objects for the purposes of interaction. For example, an item could display a schematic of a simple electrical circuit as a single graphic, and direct the respondent to click on the region of the circuit where the current flow is lowest. In order to encode the respondent's response, hotspots could be defined over any number of regions in the circuit. The hotspot scheme could range from a coarse-grained one where only two regions are defined (i.e., the region corresponding to the correct response, and the remainder of the graphic), to increasingly fine-grained schemes which would ultimately border on treating the response as an X-Y coordinate pair.

From a content development perspective, hot spots are highly efficient, since their definition data is separated from the data that describes the underlying media content. This increases the potential for reusability, since multiple items can be created using the same media content simply by substituting different hotspot schemes. For example, a collection of assessment items on electrical circuits could reuse a single circuit schematic graphic simply by changing the superimposed hotspot scheme for each item. Since hotspots are defined in the QTI XML script as numeric data, developing and modifying them will in most cases be much faster than developing and modifying graphical content. The potential for automated item generation is also increased, as the automated system would only need to be able to produce the numeric data for hotspot regions; the more complex ability to create graphics or other media would not be required.

Two important shortcomings exist, however, in the hotspot render-type as described in the current version of the QTI data model. Currently, there is no attribute that allows for the specification of whether hotspot regions are to be visually demarcated on the onscreen display in some way, such as their borders being outlined in a distinctive color. This is a parameter that needs to be under the control of the item author, as there are cases when it will be desirable to make the respondent aware of the set of possible responses to the item, and there are cases when it will be not. For example, in the circuit schematic example described above, if a coarse-grained hotspot scheme is used wherein only the correct answer is isolated from the remainder of the graphic, obtaining the correct answer will be a trivial task if the hotspot scheme is communicated to the respondent through visual demarcation. The exclusive use of fine-grained hotspot schemes would bypass this issue, of course, but this would place an unnecessary constraint on the development process. In addition to precluding the use of overlapping hot spot regions, it would also require that all hotspot-based items be visually presented as a field of numerous hot spot regions, which may reduce the comprehensibility of the underlying images. Conversely, there will also likely be instances where an item author will want to present the item as a closed set of answer options, analogous to a multiple-choice item but utilizing spatial regions on a graphic rather than the traditional list of text-based answer options. For example, the circuit schematic could be presented with four possible answer options visually demarcated on the graphic. Therefore, an additional attribute in the render_hotspot object such as showborder = {yes,no} is suggested for inclusion. A less desirable alternative to this approach would be to render hotspots with their borders visual demarcated when the response-type is LID, and render them without borders when the response-type is X-Y coordinate data.

The second shortcoming in the hotspot specification concerns the use of this object in the context of drawing lines onscreen. The QTI data model provides an attribute ('showdraw') in the hotspot render-type to specify whether mouse clicks on hotspot regions should result in lines being drawn between the regions that had been clicked on. This is a useful function for items in such subjects as geometry and geography, where respondents can be directed to select points and draw paths between them as a component of their response. A notable omission in the specification of this attribute, however, is the ability to control whether responses are to consist of line segments with two endpoints or sets of line segments. For example, in items assessing knowledge in kinematics, respondents may have to enter their responses as vector quantities. In this case, a response consisting of two vectors will be a set of ordered data points that could be represented by the following structure: {{{10, 12}, {20, 30}}, {{5,5}, {10, 10}}}. This would correspond to a first vector with tail at (10, 12) and head at (20, 30), and a second vector with head {5,5} and tail {10,10}. Conversely, in items assessing geometrical knowledge, an item may direct the respondent to create a rectangular shape onscreen whose area is equal to 20. A possible response data set for this item could be represented by {{0,0}, {5,0}, {0,4}, {5,4}}. In order for a delivery system to be able to seamlessly handle both types of items, it will need to receive data indicating whether single or multiple joined line segments were permissible in the current item. Again, this shortcoming could be bypassed by restricting item authors to problems requiring only single lines as responses, but this would be a sizeable constraint on the range of problems that could be developed. A possible solution is to include an additional value of {Multiple} in the current enumerated set of values ({No, Yes}) for the showdraw attribute of the render_hotspot object. This would allow this object to be used in conjunction with the logical group response type (with ordered cardinality) to handle items with both single and multiple line segments.

Cardinality

By providing both the functionality to retrieve response data sets with multiple elements, and the Boolean logical operators to process these responses, the QTI data model significantly increases the scope of item types that can be developed and delivered within it. For example, response sets with multiple cardinality allow performance criteria to be specified in terms of thresholds, such as "when presented with a set of triangles, of which three are acute triangles, the respondent will be able to correctly identify at least two of the acute triangles". Assuming a set of answer options such that Response = {1,2,3,4,5,6}, where the even options correspond to acute triangles and the odd to obtuse, a response processing script to implement the given performance criterion could be:

```
<resprocessing>
<respcondition continue=”no”>
<conditionvar><or>
<varsubset respident="Answer" setmatch=”exact”>{2,4,6}</varsubset >
<varsubset respident="Answer" setmatch=”exact”>{2,4}</varsubset >
<varsubset respident="Answer" setmatch=”exact”>{4,6}</varsubset >
<varsubset respident="Answer" setmatch=”exact”>{2,6}</varsubset >
</or></conditionvar>
<displayfeedback feedbacktype="Response" linkrefid="CorrectAnswer">
</respcondition>
<respcondition continue=”no”>
<conditionvar><or>
<varsubset respident="Answer" setmatch=”exact”>{2}</varsubset >
<varsubset respident="Answer" setmatch=”exact”>{4}</varsubset >
<varsubset respident="Answer" setmatch=”exact”>{6}</varsubset >
</or></conditionvar>
<displayfeedback feedbacktype="Response" linkrefid="PartiallyCorrectAnswer">
</respcondition>
<respcondition>
<conditionvar><other respident="Answer"/></conditionvar>
<displayfeedback feedbacktype="Response" linkrefid="IncorrectAnswer">
</respcondition>
</resprocessing>
```

This script would trigger the feedback event “CorrectAnswer” for any response set that met the performance criterion, “PartiallyCorrectAnswer” for responses that identified only one correct right angle, and “IncorrectAnswer” for any other answers. As can be seen from the excerpt, the varsubset method provides a mechanism for evaluating response sets. A comparison set is defined (e.g., {2,4}), and the response set is compared against the comparison set. The nature of the comparison is specified by the setmatch attribute, which specifies whether TRUE is only returned when the match is exact, or if partial matches are permissible. Therefore, in the example script above, a comparison is TRUE only if the elements of the response set are identical to the elements in the comparison set. The sequence of the elements would not be taken into account, however, since this type of item would presumably have multiple cardinality instead of ordered.

If the setmatch attribute is set to "partial", the comparison operation is presumably different. The documentation for this attribute value is extremely limited, as it was only introduced to the QTI data model one month before its public release. Based on the information available in the current specification, an attribute value of 'partial' appears to specify that the comparison test returns TRUE if the response set partially matches (i.e., is a subset of) the comparison set. Therefore, for a response set {2,4} and a comparison set {2,4,6}, TRUE would be returned if the value was "partial", as {2,4} is a subset of {2,4,6}. FALSE would be returned if the setmatch attribute value was "exact".

Given the late addition of the setmatch attribute, it is likely that the developers of the QTI data model realized that the initial scope of functionality provided for set processing was overly limited; the setmatch attribute, however, constitutes only a minor improvement. At minimum, a means for specifying the direction of the matching process for partial matching must be added. For example, in the current version of the data model it is assumed that a "partial" setmatch will return FALSE if the response set is {2,4,6} and the comparison set is {2}. That is, there is no mechanism for determining if a response set contains a certain member. This precludes the ability to isolate parts of a given response set and define customized responses based on the occurrence of individual elements or subsets that are pedagogically meaningful. One solution would be to extend the setmatch attribute to include the specification of direction, including the ability to specify nondirectionality (e.g., TRUE would be returned regardless of which of the sets was {2} and which was {2,4,6}). A more valuable addition to set processing would be the ability to evaluate set intersections in terms of the number of elements in the intersection. This could be represented through the following syntax:

```
<varsubset respident="Answer" setmatch="intersection" minnumber="2">{2,4,6}</varsubset>
```

With the addition of the attributes minnumber and maxnumber to the varsubset method, and the extension of the setmatch attribute to include "intersection", a single comparison could thereby evaluate whether the response set met the mastery criterion. This would be a relatively simple extension to implement in the QTI data model, as the attributes "minnumber" and "maxnumber" are already in use in several other methods. Furthermore, intersection operations would enable more complex response conditions to be defined, such as in the following example, where the response set is evaluated to determine if contains at least two correct answers and one or more incorrect answers:

```
<and>
<varsubset respident="Answer" setmatch="intersection" minnumber="2">{2,4,6}</varsubset>
<varsubset respident="Answer" setmatch="intersection" minnumber="1">{1,3,5}</varsubset>
</and>
```

In the case of response sets with ordered cardinality, response processing becomes somewhat more complex. As an example, consider the following hypothetical item from a computer-based tutorial on project management:

Select the tasks that should be performed during the XXX phase of the project, and sequence them in the order in which they should be performed. Remember, not all of the tasks will be used during this phase.

Task AAA	Task DDD
Task BBB	Task EEE
Task CCC	Task FFF

If the correct response set for this item was {AAA, CCC, EEE, BBB}, there may be a large number of possible response categories that are pedagogically relevant in terms of specifying customized feedback messages or for the purposes of weighted scoring models, including but not limited to:

- Responses that omit one or more correct tasks; e.g., {AAA, CCC, BBB}
- Responses that contain only the correct tasks but with two or more tasks in an incorrect sequence; e.g., {AAA, EEE, CCC, BBB}
- Responses that omit one or more correct tasks, and use an incorrect sequence for the remaining tasks; e.g., {AAA, BBB, CCC}

- Responses that contain both correct and incorrect tasks, with differing ramifications on the sequencing of the correct tasks; e.g., {AAA, CCC, EEE, BBB, FFF}, {AAA, FFF, CCC, EEE, BBB}, {CCC, AAA, FFF, EEE, BBB}

A possible approach for implementing the functionality necessary to handle complex ordered response conditions would be to expand the "intersection" operation so that it has its own cardinality attribute. For example, to determine if the response set contained only the correct tasks, but had two or more tasks in the incorrect sequence, the following script could be used:

```
<and>
<varsubset respident="Answer" setmatch="intersection" rcardinality="multiple"
minnumber="4">{AAA,CCC,EEE,BBB}</varsubset>
<not><varsubset respident="Answer" setmatch="intersection" rcardinality="multiple"
minnumber="1">{DDD,FFF}</varsubset></not>
<not><varsubset respident="Answer" setmatch="intersection" rcardinality="ordered"
minnumber="4">{AAA,CCC,EEE,BBB}</varsubset></not>
</and>
```

In order, these condition tests would verify that the response set contained all four correct tasks, that none of the incorrect tasks were included, and that the order of the response set did not correspond exactly to the correct answer.

A second useful addition to the QTI data model would be the allowance for wildcard set elements, as demonstrated in the following script by the ".." symbol:

```
<varsubset respident="Answer" setmatch="intersection" rcardinality="ordered"
minnumber="2">{..CCC..EEE..}</varsubset>
```

This response condition would return TRUE if tasks CCC and EEE were in the response set, and CCC came before EEE. The ".." wildcard symbol would be interpreted as an open subset that could have 0 to n elements; therefore, in the above response condition, 0 to n elements could occur before CCC, between CCC and EEE, and after EEE. Checking for specific positions of set elements could be implemented through the use of commas to indicate that the wildcards are to be treated as distinct characters rather than as open ranges:

<varsubset respident="Answer" setmatch="intersection" rcardinality="ordered" minnumber="1">{..,..,EEE..}</varsubset>

This response condition would return TRUE if task EEE were the third element in the ordered response set, regardless of the number of elements (if any) that followed it.

Although implementation of the above two recommendations does not provide all the functionality necessary for complex feedback schemes involving ordered response sets, it would significantly enhance the scope of the QTI data model's processing of this type of data.

Presentation and Rendering of Answer Options in Logical Groups

Although the logical group response-type is a standard element of the current version of the QTI data model, documentation of its functionality is rather scarce, and no examples are provided of its potential implementation. This is unfortunate, as this response-type is necessary for common item forms such as matching. As discussed previously, it is presumed that the response data set for this item type will consists of a set of sets. For example, if the respondent was responsible for creating one-to-one matches between two sets of elements labeled {A,B,C,D} and {1,2,3,4}, a possible response data set would be: {{A,1}, {B,4}, {C,2}, {D,3}}. Response conditions could then evaluate whether the response data contained certain subsets, e.g. {A,1}.

The lack of examples for this response-type in the current version of the QTI data model is understandable, since there are key omissions from the articulation of this response-type that will preclude its correct operation. Chief amongst these is an inability to specify membership in a subset when defining response alternatives. For example, in an item that directs the candidate to match parametric tests in statistical analysis with their nonparametric equivalents, the current version of the QTI data model would assumedly utilize the following representation for the response options:

<presentation><response_grp ident="Q1">

<material>Match the parametric tests with their nonparametric equivalents.</material>

<render_choice>

<response_label ident="A"><material> Correlated samples t test </material></response_label>

<response_label ident="B"><material> Student's t test </material></response_label>

<response_label ident="C"><material> Oneway ANOVA </material></response_label>

<response_label ident="D"><material> Within-Groups ANOVA </material></response_label>

<response_label ident="1"><material> Friedman ANOVA </material></response_label>

<response_label ident="2"><material> Kruskal-Wallis H test </material></response_label>

<response_label ident="3"><material> Wilcoxon signed-ranks t test </material></response_label>

<response_label ident="4"><material> Mann-Whitney U test </material></response_label>

</render_choice>

</response_grp>

</presentation>

Although the labeling of the answer options as {A, B, C, D, 1, 2, 3, 4} would likely be sufficient for a human reader to understand that there are two distinct subsets within the set of response alternatives, there is no attribute here that explicitly defines this aspect in a manner tractable for computer-based processing. This could lead to numerous problems. For example, in terms of presentation of the item, the delivery system would be just as correct in its interpretation of the item data if it listed the answers in a single column as it would be if it listed them in two columns of four elements each. Item authors would have no way of specifying when subset membership should be communicated to respondents (i.e., the item indicates which tests are parametric and which are nonparametric) and when this information should be hidden. This information will likely have important ramifications for the difficulty of the item. An additional requirement is that randomized shuffling of response alternatives must respect subset memberships, in that shuffling should occur within subsets but not across them.

The definition of explicit subsets is also required for adequately specifying the functionality of the logical group response-type. The lack of this information precludes the delivery system from enforcing any restrictions on the matching relationships made by the respondent. In the above example, item authors would have no way of specifying whether matches within a subset, such as {A,B}, were permissible, since there is no means provided for the definition of subsets in the item. Ideally, a mechanism would also be incorporated for specifying permissible matches between subsets as one-to-one, many-to-one, one-to-many, or many-to-many.

Specification of Fill-in-the-Box Prompts

One aspect of presentation that the QTI data model should not extend into is the specification of the rectangular size of the text entry box for relatively lengthy responses in fill-in-the-blank items such as short-answer and essay items. When the expected number of characters in a response is large relative to the smallest display size of the delivery platforms served by the assessment system, each delivery system must preserve the ability to reconfigure the text entry region to best utilize its available screen space. Currently, the QTI data model allows item authors to specify the implementation of the text entry region in terms of number of rows and columns, as well as the maximum number of characters allowed in the response. Although the area of a text response region has pedagogical significance (e.g., implicitly cueing the respondent regarding the expected length of an 'acceptable' response), the specific division of this area into rows and columns does not. Therefore, the ability to specify rows and columns should be eliminated.

Moveable Objects as a Render Type

In order to allow individual organizations to extend the QTI data model for context-specific functionality, numerous opportunities for defining proprietary data elements are included within the specification. Although the inclusion of proprietary data elements could be seen as being at odds with the goal of defining a standardized data model, the provision of a formalized mechanism for doing so increases the extensibility of the data model without necessarily compromising its ability to serve as a standard. Provided the additional information included in the extended data model does not prohibit its successful execution in a QTI-compliant system, the objective of interoperability can be preserved. For example, an organization could define custom response processing data elements within a set of items that override those native to the QTI data model when the delivery system is capable of implementing the custom elements. If the items also include sufficient information using the standard data elements for the item to function adequately in the absence of the custom functionality, the items will remain usable when the delivery system is only compliant at the level of the core QTI data model.

Custom data elements also enable the functionality of the core data model to be extended and field tested between formal version releases. An interesting case occurs in the current version of the QTI data model, where the specification developers themselves have included an additional render type through the extension mechanism, with the caveat that it is not an official element of the data model. This render type, 'ims_render_object', enables the definition of a set of media objects as moveable within a single dimension. For example, a list of steps in hypothesis testing in statistical analysis could be displayed in a vertical list, with the candidate responsible for organizing the steps in the correct sequence by moving each step up or down in the list. Assuming that the item was an ordered LID response-type, the response data set would be in the form of a list of response labels whose order reflected the horizontal sequence in the respondent's response. In traditional print-based versions of items of this nature, the respondent generally writes in a number beside each step to signify its place in the correct sequence. This format for response entry could be replicated in the core QTI data model, but the render_object extension allows item authors to take advantage of the increased level of manipulation possible in a computer-based WIMP environment.

In the next version of the data model, the render_object extension will need to be formalized and incorporated into the standard. An important issue in this respect will be the application of this extension to X-Y coordinate response types. This will allow the QTI data model to implement items where the two-dimensional coordinates of moveable objects are processed as responses. An example of an item of this nature would be a bar graph displayed onscreen, where the respondent would have to drag and place text labels such as 'mode', 'abscissa', and 'ordinate' onto their correct locations on the graphic. Currently, the QTI data model purports to handle this type of item using hotspots in conjunction with X-Y coordinates, but the omission of a means for specifying whether hotspots are moveable or not will preclude this framework from functioning properly in an actual delivery system. Hotspots should be maintained as immobile definitions of regions relative to a fixed graphic, and the render_object type should be used when moveable media objects are required. In the response data set for a render_object item, each object needs to have its own identification label and X-Y coordinate pair, enabling the definition of conditions in response processing for the evaluation of the individual objects in relation to regions on the bar graph graphic.

Full implementation of the render_object type will also require a clearer specification of the manner in which the QTI data model handles overlapping media objects. Currently, the position of media objects is specified through x-y coordinate pairs. There is no explicit representation of Z layers, that is, there is no specific provision for dividing the interface into layers for the purposes of determining which media object has priority when there is an overlap between multiple objects. The need for Z layer information was likely discovered during the development of the original specification, as there is a statement that "in the case of overlapping images the order of precedence is defined by the order of the response_label elements – the first declared has the highest precedence" (Smythe & Shephed, 2000, p. 22). This statement, however, fails to clarify the relationship between the media object(s) that serves as the underlying context for moveable objects (e.g., the bar graph in the example described above) and the moveable objects themselves.

To resolve this issue, media objects require an additional attribute that allows for the specification of Z layer. In some cases, this would also allow for the increased reusability of media objects across items. For example, in the context of electrical circuits, a basic 'circuit board' graphic could be produced once and used as the background for additional graphics. These additional graphics could consist of individual circuit components such as resistors, wires, batteries, and so forth. The background graphic would occupy the 'lowest' Z position, and the component graphics would occupy a second 'middle' Z position, which would mean that the component graphics are displayed 'over' the background circuit board. Thus, rather than producing a custom circuit graphic for each individual item, these component graphics could be reused across items by reconfiguring them using X and Y coordinate information. Respondents could be directed to place moveable objects such as ohmmeters or voltmeters on the graphic schematic, and these objects would occupy a third 'top' Z layer. Once the basic set of media objects was created, production and revision time for these items would be proportional to the time required to enter or change X-Y coordinate data rather than the time required to create new media objects. In addition, the ability to automate item generation for items containing graphical media objects would be greatly increased.

Incorporation of Text Through URIs

To realize the potential impact of the QTI data model on the efficiency of instructional content development and maintenance tasks, the separation of data from logic must be made wherever possible.

The QTI data model provides a highly useful mechanism for achieving this by allowing Uniform Resource Identifiers (URI) to be used in place of incorporating the content within an ASI unit. The URI provides a means for identifying the access instructions for an object within a given name space, such as the WWW (Berners-Lee, 1994). This allows, for example, for images and other media objects to be used as part of an item, as demonstrated in the script excerpt below:

```
<material><mattext>Based on the bar graph shown, answer the question below.</mattext></material>
<material><matimage imagtype="image/gif" uri="http://www.sassi.com/testbank/images/bargraph1.gif">
</matimage></material>
```

Whereas the text ("Based on the bar graph shown...") in the above example is contained within the logical specification of the item in the QTI data model, the graphic content is incorporated through a pointer to the media object's location, which in this case is a WWW URL ("http://www.sassi.com/testbank/images/bargraph1.gif"). This enables the QTI data model to incorporate any media object which can be retrieved over a network, including static graphics, animations, video and audio, as well as Java applets and other software applications. Although text can be referenced as a media object through a URI as well, it is invariably incorporated within the XML script in print-based examples of the QTI data model for the purposes of making the scripts more easily comprehensible to readers. This can unfortunately imply that the best practice in actual implementations of the data model is to do likewise, when the opposite will be true in circumstances where generic rubrics and feedback messages are used. For example, a large item bank may include thousands of multiple-choice questions, each with the text rubric "Select the best answer from the list below." If, at some point in time, it was decided that the text for this rubric should be changed to "Select the single best answer from the list below.", the time to effect this change would be dependent on whether this text was physically contained in each individual item. If it was, then the revision time would be proportional to the number of items to be revised, which would likely be substantial in the case of a large item bank. Although an automated search and replace process could be used in this case, it is not a failsafe approach in the context of an open distributed system, since minor variations between instances of the media element would not be identified. Conversely, if each item contained a URI as a pointer to a single file containing the text string, only the contents of this file would have to be changed, and the revision would automatically propagate throughout the item bank. Using URIs for text also facilitates the enforcement of organizational standards for generic text elements, which can be an important issue in managing large projects where numerous individuals have authoring responsibilities (Bharwaj, Chandran, O'Neal & Gibbons, 1998). The human readability of the XML script can be maintained by encouraging item authors to include explanatory comments within the script using qticomment data, where the consequences of individual authors' variability are much less critical.

Expansion of Views to Item Responses

The QTI data model also provides an excellent vehicle for including explanatory documentation to be displayed at run-time. By specifying different values for the view attribute of certain objects, the item author can control who sees the documentation attached to these objects:

```
<itemfeedback ident="Feed1" view="candidate">
<material><mattext>Your answer is incorrect.</mattext></material></itemfeedback>
<itemfeedback ident="Feed1" view="tutor">
<material><mattext>This response may indicate that the candidate does not understand the difference
between Type I and Type II errors in hypothesis testing.</mattext></material></itemfeedback>
<itemfeedback ident="Feed1" view="psychometrician">
<material><mattext>This incorrect answer option is selected by 19% of
respondents.<mattext></material></itemfeedback>
```

Thus, when reviewing items from an item bank through a QTI-compliant assessment system, tutors and psychometricians could interact with the items while simultaneously viewing role-specific commentary. In addition to these roles, the QTI data model also provides views for scorers, proctors, administrators, and item authors. A minor limitation is that views can only be associated with certain types of data, such as rubrics and item feedback. It may be useful to allow the specification of view with all presented material, including item stems and answer options.

Modular Architecture as Implemented by the QTI Specification

The current version of the QTI specification is a promising beginning. In projects where a large amount of computer-based assessment content must be developed and/or managed, it can serve as the blueprint for a viable system architecture. In this final section of the thesis, the manner in which the specification implements a modular approach to instructional design within distributed learning environments is discussed.

Modularity within an Item

As discussed previously, a given computer-based assessment item articulated through the QTI specification can be seen as being divided into three primary components:

- A component responsible for presenting the item's stimulus material, such as the item stem and rubric, and the mechanism for entering responses, such as a list of response alternatives.
- A component responsible for receiving a respondent's responses to the item, categorizing these responses in terms of their pedagogical relevance, and using these categories to trigger subsequent events dependent on the nature of the response.
- A component responsible for specifying feedback content.

By considering an individual item as a distinct system, these three components can be considered as individual modules that exchange messages with one another as well as performing internal operations. This division is reflected within the XML script through the three main top-level item tags: <presentation>; <resprocessing>; and <itemfeedback>. The interface between the first module (item presentation) and the second module (response processing) is defined by the response data, and therefore also by the action of a respondent entering a response. To illustrate the effect of this partition, a situation wherein two designers were working on the same item could be considered. In this context, the designer working on the first module would be free to alter any parameter of its design save for those that changed an aspect of the response data that would be received by the second module. For example, this designer could change the wording of the item stem without necessarily alerting the other designer, but not the identification label of one of the response alternatives. Similarly, the text of a response alternative could be freely modified in the first module provided it did not impact any of the categorization operations performed in the second module; for example, if the modification changed a response alternative from a correct answer into a partially correct answer, changes would also have to be made in the second module.

In the case of the second module, response data is received and translated into pedagogically-salient conditions such as a 'correct answer'. The occurrence of one of these conditions is used to trigger events associated with the condition, such as the modification of a variable or the presentation of a feedback message to the respondent. It is interesting to note that one type of event, variable manipulation, is handled internally within this second module, while the other, feedback delivery, is handled in a separate module. Further modular decomposition in the standard could be considered by limiting the operational role of the second module to translation of raw response data into response conditions, followed by the broadcasting of these response conditions to separate modules responsible for variable manipulation or feedback delivery. In the current version of the standard, however, only feedback delivery is handled by a separate module. This allows item developers to freely modify the contents of feedback messages provided that the relationships between a given feedback event and its associated response conditions do not change. The information contained in these relationships therefore constitutes the interface between the second and third modules, in a similar fashion to the interface between the first and second.

Thus, at the item level, the QTI data model decomposes an item into three distinct components, and formalizes their interactions. This modularization can serve to reduce the complexity associated with designing a computer-based assessment item's structure and functionality.

Three-Tiered Architectural Model

Another perspective from which to view the modularity inherent in the QTI data model is to consider it as a module unto itself, in terms of its articulation of the logic of an assessment instrument. Then, the module that is responsible for converting the information in the QTI specification into the desired structure and functionality on a target delivery platform could be considered as the 'presentation' module. A third, separate 'data' module would consist of the databases containing the raw media elements that are configured and manipulated based on the information in the QTI specification. Thus, the information expressed through the QTI data model functions as an interstitial module, or middle tier, in a system divided into data, logic and presentation components. This middle tier functions as a bidirectional translator, specifying the generic elements of an assessment's presentation in a manner that is usable across delivery platforms, and understanding generic categories of assessment content in a manner that is usable across assessment contexts. It also codifies the possible actions of both the respondent and the system during the operation of an assessment item.

Management of Multiple Instructional Designers on a Single Project

There are two primary means by which the QTI specification can support the management of a design project involving multiple designers. By formalizing the components of a computer-based assessment item, the allocation and scheduling of design tasks can occur in a more explicit manner than is possible in the absence of such a representation. For example, initial design efforts may focus on the core functionality required by a majority of item types, followed by a preliminary population of the databases with assessment content for the purposes of testing the system. After the tests are completed, designers could be divided into two groups to concurrently develop extensions to the system's functionality while the databases are further populated to reach the desired level of content for the first system release. By erecting a partition between these two design teams, work can proceed in parallel without incurring the overhead of identifying and resolving design dependencies. The system-wide blueprint provided by the specification also allows progress and risk exposure to be more easily measured and monitored.

A second means by which design resources can be better managed through a data model such as the QTI specification is through the partitioning of the system into open and closed modules. For example, a small team of senior instructional designers could define an assessment system's overall structure and functionality, and develop standardized content elements such as generic feedback messages. These aspects of the system can then be closed and left in 'read-only' state, with junior instructional designers responsible for the completion of the assessment system within the architecture developed by the senior designers.

An Explicit Interface within a Multidisciplinary Design Team

The QTI data standard can provide a framework for managing the interactions between members of a multidisciplinary design team. The most obvious case here would be between instructional designers and software developers, in that the QTI data standard would function as an interface between these two roles. The software developers' responsibility would be to enact the structure and functionality expressed by the standard in actual target computer systems. Conversely, the instructional designers' responsibility would be to express the intended pedagogical design within the conceptual model articulated through the QTI data standard. This division of roles may seem commonsensical, but in achieving a smooth workflow across this division it is useful to have an explicit, documented, and tested manifestation of the interface that codifies it.

Representing a Key Facet of the Design Space

Speaking in general terms, the goal of a researcher could be considered to be the derivation of a description of the structure and function of a given phenomenon, whereas the goal of a designer could be considered to be the creation of artifacts that change a given aspect or state of the world into a more desirable one. It would seem appropriate, then, for an applied domain such as educational technology to train practitioners in the mold of 'researchers-designers', in a similar vein to the 'scientist-practitioner' model espoused in clinical psychology. Simon (1996) has argued, however, that higher/professional education has increasingly centered on research at the expense of practice over the course of the 20th century, and on description at the expense of prescription, even within domains that are ostensibly centered on the latter, such as engineering, medicine, and business. Although an argument could be made that a certain threshold of adequate description must precede prescription, and thus these domains' focus on basic research may be a sign that they are still in a nascent period of knowledge development, a different position on the root cause for this trend is taken by Simon: the desire in academia for the "academic respectability" connotated by "intellectually tough, analytic, formalizable, and teachable" subject matter, rather than the "intellectually soft, intuitive, informal, and cookbooky" subject matter of design (Simon, 1996, p.112). To counterbalance this situation, he proposed seven topics in the theory of design that should be included in curricula intended for practitioners, the last of which was the representation of design problems.

In the domain of computer-based assessment systems, the QTI specification serves as an example of the kind of representation that can assist instructional designers in formalizing and refining their understanding of their design space. As Simon (1996) stresses, the influence of these representations on the actual course of a design project should not be underestimated. The existence of a usable and fleshed out model of the structure and functionality of a computer-based assessment instrument at the outset of a design project helps to ensure that this facet of instructional design can be tractably approached within the constraints of the typical development effort. Conversely, Postman (1992) reminds us of the adage, 'to a man with a hammer, everything looks like a nail', in order to drive home the point that in the age of the ubiquitous computer, everything can look like data waiting to be processed. A cautionary tale in this regard can be elicited from architectural education, wherein at least one educator has noted that the introduction of CAD tools, although a boon to architects, has also resulted in the unfortunate tendency for junior architects to design buildings that look great from a 'bird's eye view' rather than from ground level, after spending too much time optimizing the aesthetics of their virtual buildings from the former perspective within the CAD environment (Robbins, 1994; see also Rowe, 1987). Thus, it would be a mistake to position the QTI specification as the sole tool necessary to adequately model a computer-based assessment environment during design. Doing so might easily blind designers to the facets better captured by other design tools such as static two-dimensional storyboards and functional prototypes, let alone those facets related to task and learner analyses, and the derivation of learning objectives. Instead, a data model such as the QTI specification should be considered as a necessary and valuable addition to the portfolio of techniques and tools that instructional designers bring to a design task, one that allows designers to modularize a computer-based system from a number of perspectives and analyze the information that flows between these modules.

An Interface Between Theory and Design

A final perspective from which to consider the potential impact of an information model such as the QTI standard is in the context of the relationship between descriptive theory and prescriptive design in educational technology. A good starting point here is the early concept of aptitude-treatment interactions (ATI): that is, that the optimal instructional treatment in a given context is dependent on the aptitude of the learner in that context (Cronbach & Snow, 1977). Based on this premise, research should arguably investigate the aspects of instructional treatments that interact with aptitude, in addition to the more general qualities of effective treatments that lead to optimal results across implementation contexts. In furthering our understanding, the ATI concept need also be extended in several ways: through the incorporation of other traits or attributes of the learning context in addition to aptitude, such as the nature of the subject matter (e.g., Jonassen & Grabowski's (1993) concept of Content by Treatment Interactions); through the incorporation of state variables that are measured during the course of an instructional treatment, and the adaptation of the treatment during its implementation rather than solely prior to it; and through the more precise articulation of both the variables to be measured and the differential actions that are taken based on the results of these measurements. This latter aspect is the hallmark of an explicit and publicly-defensible design rationale.

In this regard, the QTI specification can serve as a model of how a broad set of treatment or design variables might be formalized and operationalized, since here a large share of a designer's possible actions in developing a computer-based assessment system are articulated through an explicit set of attributes, parameter values and relationships. To fully realize the potential impact of this type of formalized model, future research will need to focus on deriving design prescriptions that link the QTI data model to both state and trait variables in the learning context through theoretical models that are empirically validated. Hopefully, the increased efficiency with which modular architectures can be used to develop, operate and maintain distributed learning environments will also drive a significant reduction in the resource requirements associated with collecting and analyzing data from large samples, thereby paying dividends to researchers as well as designers, and accelerating the specification's own evolution.

References

Abrami, P. C., Cholmsky, P. & Gordon, R. (2001). Statistical analysis for the social sciences: An interactive approach. Needham Heights, MA: Allyn & Bacon.

Advanced Distributed Learning Initiative. (2000). Shareable courseware object reference model. [online]. Available: http://www.adlnet.org

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. The Journal of the Learning Sciences, 4 (2), 167-207.

Anderson, T., McArthur, D., Griffin, S. & Wason, T. (1999). IMS Meta-data Best Practice and Implementation Guide. Wilmington, DE: IMS Global Learning Consortium. Available: http://www.imsproject.org

Anderson, T. (2000). IMS content packaging best practice guide. Burlington, MA: IMS Global Learning Consortium. Available: http://www.imsproject.org

Andrews, D. H. & Goodson, L. A. (1980). A comparative analysis of models of instructional design. Journal of Instructional Development, 3 (4), 2-16.

Armstrong, P., & Tomes, A. (1996). Art and accountability: the languages of design and managerial control. Accounting, Auditing and Accountability Journal, 9 (5), 114-125.

Ausubel, D. P. (1968). Educational psychology: A cognitive view. New York: Holt, Reinhart, and Winston.

Baldwin, C.Y., & Clark, K.B. (1997). Managing in an age of modularity. Harvard Business Review, September-October, 84-93.

Berkun, S. (1999). How to avoid foolish consistency. Seattle, WA: Microsoft Corp. Available: http://msdn.microsoft.com/library/welcome/dsmsdn/hfactor8_5.htm

Berners-Lee, T. (1994). Universal resource identifiers in WWW: A unifying syntax for the expression of names and addresses of objects on the network as used in the World-Wide Web (RFC1630). Geneva, Switzerland: CERN. Available at: http://www.faqs.org/rfcs/rfc1630.html

Bharwaj, K. K., Chandran, P., O'Neal, A. F., & Gibbons, A. S. (1998). Large scale development of learning materials: critical issues. Educational Technology Research and Development, 46 (2), 100-110.

Birnbaum, A. (1968). Some latent trait models. In F.M. Lord & M.R. Novick, (Eds.), Statistical theories of mental test scores. Reading, MA: Addison-Wesley.

Bloom, B.S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. Educational Researcher, 13, 4-16.

Blumenthal, R., Meiskey, L., Dooley, S., & Sparks, R. (1996, June). Reducing Development Costs with Intelligent Tutoring System Shells. Paper presented at the ITS'96 Workshop on Architectures and Methods for Designing Cost-Effective and Reusable ITSs, Montréal, Canada. Available: http://advlearn.lrdc.pitt.edu/its-arch/papers/blumenthal.html

Buck, G. H. (1989). Teaching machines and teaching aids in the ancient world. McGill Journal of Education, 24 (1), 31-54.

Bush, V. (1945). As we may think. The Atlantic Monthly, July, 101-108.

Cambre, M.A. (1981). Historical overview of formative evaluation of instructional media products. Educational Communication and Technology, 29, 3-25.

Carroll, J. M. (1990). The Nurnberg Funnel: Designing minimalist instruction for practical computer skill. Cambridge, MA: The MIT Press.

Chapman, B. L. (1994). Enhancing interactivity and productivity through object-oriented authoring: An instructional designer's perspective. Journal of Interactive Instruction Development, 7 (2), 3-11.

Clark, K., & Dwyer, F. (1998). Effect of different types of computer-assisted feedback strategies on achievement and response confidence. International Journal of Instructional Media, 25 (1), 55-63.

Clark, R. E. (1983). Reconsidering research on learning from media. Review of Educational Research, 53 (4), 445-459.

Clark, R. E. (1984). A reply to Petkovich and Tennyson. Educational Communications and Technology Journal, 32, 238-241.

Clark, R. E. (1994). Media will never influence learning. Educational Technology Research and Development, 42 (2), 21-29.

Cholmsky, P. (2000). Design decisions in the development of modular instructional content: Describing the transition from local to global learning systems. In J. Bourdeau & R. Heller (Eds.), Proceedings of ED-MEDIA 2000 - World Conference on Educational Multimedia, Hypermedia and Telecommunications (pp. 1576-1577). Phoenix, AZ : Association for the Advancement of Computing in Education.

Cronbach, L. & Snow, R. (1977). Aptitudes and instructional methods: A handbook for research on interactions. New York: Irvington.

Cusumano, M. A., & Yoffie, D. B. (1998). Competing on Internet time: Lessons from Netscape and its battle with Microsoft. New York, NY: Scribner.

Derry, S.J., & Lajoie, S.P. (1993). A middle camp for (un)intelligent instructional computing: An introduction. In S.P. Lajoie & S.J. Derry (Eds.), Computers as cognitive tools (pp.1-11). Hillsdale, NJ: Erlbaum.

Dick, W. & Carey, L. (1990). The systematic design of instruction (3rd edition). Glenview, IL: Scott, Foresman.

Ebel, R. L., & Frisbie, D. A. (1986). Essentials of educational measurement (4th edition). Englewood Cliffs, NJ: Prentice-Hall.

Ehrmann, S. C. (1995). Asking the right question: What does research tell us about technology and higher learning? Change, the Magazine of Higher Learning, 27 (2), 20-27.

Embretson, S. E. (1999). Generating items during testing: Psychometric issues and models. Psychometrica, 64 (4), 407-433.

Ertmer, P. A., & Newby, T. J. (1993). Behaviorism, cognitivism, constructivism: Comparing critical features from an instructional design perspective. Performance Improvement Quarterly, 6 (4), 50-70.

Farrance, F., & Tonkel, J. (1999). Learning technology systems architecture. New York, NY: Edutool.

Gagne, R. M., & Briggs, L. J. (1979). Principles of instructional design. New York, NY: Holt, Rinehart & Wilson.

Garrison, J. W. (1986). Some principles of postpositivistic philosophy of science. Educational Researcher, 15 (9), 12-18.

Gentner, D., & Gentner, D. R. (1983). Flowing waters or teeming crowds: Mental models of electricity. In D. Gentner and A. L. Stevens (Eds.), Mental models (pp. 75-98). Hillsdale, NJ: Erlbaum.

Gibbons, A. S. , Bharwaj, K. K., & Richards, R. (1998). The single-parse method of design for problem-based instruction. Educational Technology Research & Development, 45 (2), 110-116.

Glasersfeld, E. von. (1995). Radical constructivism: a way of knowing and learning. Washington, DC: Falmer Press.

Graves, W. H. (1999). The instructional management systems cooperative: converting random acts of progress into global progress. Educom Review, 34 (6), 32-36.

Green, B. F., Bock, R. D., Humphreys, L. G., Linn, R. L., & Reckase, M. D. (1984). Technical guidelines for assessing computerized adaptive tests. Journal of Educational Measurement, 21 (4), 347-360.

Gustafson, K. L., & Branch, R. M. (1997a). Survey of instructional development models (3rd ed.). Syracuse, NY: ERIC Clearinghouse on Information & Technology.

Gustafson, K. L., & Branch, R. M. (1997b). Revisioning models of instructional development. Educational Technology Research & Development, 45 (3), 73-89.

Hackos, J. T. (1994). Managing your documentation projects. New York, NY: John Wiley & Sons.

Haladyna, T. M. (1997). Writing test items to evaluate higher order thinking. Needham Heights, MA: Allyn & Bacon.

Hannafin, M. J., Hannafin, K. M., Land, S. M., & Oliver, K. (1997). Grounded practice and the design of constructivist learning environments. Educational Technology Research & Development, 45 (3), 101-117.

Heckel, P. (1991). The elements of friendly software design. (2nd ed.). Alameda, CA: Sybex Inc.

Hofstadter, D. R. (1985). Metamagical themas : questing for the essence of mind and pattern. New York : Basic Books.

Jacobson, M. J., & Spiro, R. J. (1995). Hypertext learning environments, cognitive flexibility, and the transfer of complex knowledge: An empirical investigation. Journal of Educational Computing Research, 12 (5), 301-333.

Jonassen, D. H., & Grabowski, B. L. (1993). Handbook of individual difference, learning, and instruction. Hillsdale, NJ: Lawrence Erlbaum.

Jonassen, D. H. (1994). Technology as cognitive tools: Learners as designers. [online]. Available: http://itech1.coe.uga.edu/itforum/paper1/paper1.html

Jonsen, A. R., & Toulmin, S. E. (1988). Abuse of casuistry: A history of moral reasoning. Berkeley, CA: University of California Press.

Klein, J. D. (1997). ETR&D-Development: An analysis of content and survey of future direction. Educational Technology Research & Development, 45 (3), 57-62.

Kolodner, J. L., & Leake, D. B. (1996). A tutorial introduction to case-based reasoning. In D. B. Leake (Ed.), Case-based reasoning: Experiences, lessons & future directions (pp. 31-65). Cambridge, MA: MIT Press.

Kozma, R. B. (1994). Will media influence learning? Reframing the debate. Educational Technology Research and Development, 42 (2), 7-19.

Kozma, R. (2000). Reflections on the state of educational technology research and development. Educational Technology Research & Development, 48 (1), 5-15.

Lajoie, S. P., & Lesgold, A. (1989). Apprenticeship training in the workplace: Computer coached practice environment as a new form of apprenticeship. Machine-Mediated Learning, 3, 7-28.

Lawson, B. (1980). How designers think. Westfield, NJ: Eastview.

Laurel, B. (1992). On finger-flying and other faulty notions. In L .Jacobson (Ed.), Cyberarts: Exploring art & technology (pp. 286-291). San Francisco, CA: Miller Freeman.

Lesgold, A. (1998). On evaluating educational innovations. Working paper from Learning Research and Development Center, University of Pittsburgh. Available: http://alan.lrdc.pitt.edu/ostp2.pdf

Linden, W. J. van der, & Hambleton, R. K. (1997). Handbook of modern item response theory. New York, NY: Springer-Verlag.

Mager, R. (1962). Preparing instructional objectives. Palo Alto, CA: Fearon.

Maher, J. H., & Ingram, A. L. (1989, February). Software engineering and ISD: Similarities, complementaries, and lessons to share. Paper presented at the meeting of the Association for Educational Communications and Technology, Dallas TX.

Marchionini, G. (1995). Information seeking in electronic environments. Cambridge, UK: Cambridge University Press.

Mark, M. A., & Greer, J. E. (1993). Evaluation methodologies for intelligent tutoring systems. Journal of Artificial Intelligence in Education, 4 (2/3), 129-153.

Mayer, R. E. (1997). Multimedia learning: Are we asking the right questions? Educational Psychologist, 32 (1), p. 1-19.

McArthur, D. (1992). Some possible futures for artificial intelligence in mathematics education. Invited talk given at the Fifth International Conference on Technology in Collegiate Mathematics, Chicago, November, 1992. Available: http://www.rand.org/hot/mcarthur/papers/future.html

McLuhan, M. (1964). Understanding media: The extensions of man. New York, NY: McGraw-Hill.

Merrill, M. D. & ID2 Research Team (1996). Instructional transaction theory: Instructional design based on knowledge objects . Educational Technology, 36 (3), 30-37.

Miller, S. I., & Fredericks, M. (1991). Postpositivistic assumptions and educational research: Another view. Educational Researcher, 20 (4), 2-8.

Mohr, L. B. (1992). Impact analysis for program evaluation. Newbury Park, CA: SAGE Publications.

Mory, E. H. (1992). The use of informational feedback in instruction: Implications for future research. Educational Technology Research & Development, 40 (3), 5-20.

Murray, T. (1998). Authoring knowledge based tutors: Tools for content, instructional strategy, student model, and interface design. Journal of the Learning Sciences, 7 (1), 5-64.

National Science Foundation (2000). Interagency Education Research Initiative Program Solicitation. Available: http://www.nsf.gov/pubs/2000/nsf0074/nsf0074.txt

Newell, A. (1973). You can't play 20 questions with nature and win: Projective comments on the papers in this symposium. In W. G. Chase (Ed.), Visual information processing (pp. 283-308). New York: Academic Press.

Nielsen, J. (1993). Usability engineering. New York, NY: AP Professional.

O'Neal, A. F. (1996). Authoring productivity environments. [online]. Available: http://www.whidbey.com/frodo/apt.htm

Orey, M., Rieber, L., King, J., & Matzko, M. (2000). The studio: Curriculum reform in an instructional technology graduate program. Paper presented at the meeting of the American Educational Research Association, New Orleans, LA.

Osterlind, S. J. (1998). Constructing test items: Multiple-choice, constructed-response, performance, and other formats. Boston, MA: Kluwar Academic.

Patrick, J., Gregov, A., & Halliday, P. (2000). Analyzing and training task analysis. Instructional Science, 28, 51-79.

Perry, W. G. (1970). Forms of intellectual and ethical development in the college years: A scheme. New York, NY: Holt, Rinehart, and Winston.

Postman, N. (1992). Technopoly: the surrender of culture to technology. New York, NY: Knopf.

Rehder, B., Schreiner, M. E., Wolfe, M. B., Laham, D., Landauer, T. K., & Kintsch, W. (1998). Using Latent Semantic Analysis to assess knowledge: Some technical considerations. Discourse Processes, 25, 337-354.

Reiser, R. A., & Ely, D. P. (1997). The field of educational technology as reflected through its definitions. Educational Technology Research & Development, 45 (3), 63-72.

Reye, J. (1996, June). Reusability via formal modelling. Paper presented at the ITS'96 Workshop on Architectures and Methods for Designing Cost-Effective and Reusable ITSs, Montréal, Canada. Available: http://advlearn.lrdc.pitt.edu/its-arch/papers/koedinger.html

Reinertsen, D. G. (1997). Managing the design factory: a product developer's toolkit. New York,NY: Free Press.

Richey, R. C. (1997). Research on instructional development. Educational Technology Research & Development, 45 (3), 91-100.

Richey, R. C. (1998). The pursuit of usable knowledge in instructional technology. Educational Technology Research & Development, 46 (4), 7-22.

Ritter, S. & Koedinger, K. R. (1996, June). Towards Lightweight Tutoring Agents. Paper presented at the ITS'96 Workshop on Architectures and Methods for Designing Cost-Effective and Reusable ITSs, Montréal, Canada. Available: http://advlearn.lrdc.pitt.edu/its-arch/papers/koedinger.html

Robbins, E. (1994). Why architects draw. Cambridge, MA: MIT Press.

Roid, G. H., & Haladyna, T. M. (1982). A technology for test-item writing. New York, NY: Academic Press.

Rose, N., & Van Horn, C. (1956). Theory and application of preproduction testing. AV Communication Review, 4, 21-30.

Roskam, E. E. (1997). Models for speed and time-limit tests. In W. J. van der Linden & R. K. Hambleton (Eds.), Handbook of modern item response theory. New York, NY: Springer-Verlag. 187-208.

Rowe, P. G. (1987). Design thinking. Cambridge, MA: MIT Press.

Rowley, K. (1996, June). Increasing the Effectiveness of ITS Research and Development. Paper presented at the ITS'96 Workshop on Architectures and Methods for Designing Cost-Effective and Reusable ITSs, Montréal, Canada. Available: http://advlearn.lrdc.pitt.edu/its-arch/papers/rowley.html

Rudner, L. M. (1998). An on-line, interactive, computer adaptive testing tutorial. [online]. College Park, MD: ERIC Clearinghouse on Assessment and Evaluation. Available: http://ericae.net/scripts/cat/catdemo.htm

Russell, T. L. (1999). The no significant difference phenomenon. [online]. Available: http://nova.teleeducation.nb.ca/nosignificantdifference/

Saettler, P. (1990). The evolution of American instructional technology. Englewood, CO: Libraries Unlimited.

Salomon, G. (1991). Transcending the qualitative-quantitative debate: The analytic and systemic approaches to educational research. Educational Researcher, 20 (6), 10-18.

Samejima, F. (1997). Graded response model. In W. J. van der Linden & R. K. Hambleton (Eds.), Handbook of modern item response theory (pp. 85-100). New York, NY: Springer-Verlag.

Schön, D. A. (1993). The reflective practitioner: how professionals think in action. New York, NY: Basic Books.

Seels, B. B., & Richey, R. C. (1994). Instructional technology: The definition and domains of the field. Washington, DC: Association for Educational Communications and Technology.

Shapiro, A. M., & McNamara, D. S. (2000). The use of latent semantic analysis as a tool for the quantitative assessment of understanding and knowledge. Journal of Educational Computing Research, 22 (1), 1-36.

Shaw, S. G. (1990). An examination of the arguments against the naturalistic paradigm of research in educational technology and their implications for current research practices. Unpublished doctoral dissertation, Concordia University, Montreal, Canada.

Simon, H. A. (1996). The sciences of the artificial (3rd edition). Cambridge, MA: MIT Press.

Skinner, B. F. (1954). The science of learning and the art of teaching. Harvard Educational Review, 24, 86-97.

Skinner, B. F. (1958/1996). Teaching machines. In D. P. Ely, & T. Plomp (Eds.), Classic writings on instructional technology (pp. 211-227). Englewood, CO: Libraries Unlimited.

Sleeman, D. H., & Brown, J. S. (1982). Intelligent tutoring systems. New York, NY: Academic Press.

Smith, P. L., & Ragan, T. J. (1993). Instructional design. New York, NY: MacMillan.

Smythe, C., & Shephed, E. (2000). IMS Question & Test Interoperability Information Model Specification. Burlington, MA: IMS Global Learning Consortium. Available: http://www.imsproject.org

Snow, C. P. (1969). Two cultures and the scientific revolution. London, UK: Cambridge University Press.

Spray, J. A. (1997). Multiple-attempt, single-item response models. In W. J. van der Linden & R. K. Hambleton (Eds.), Handbook of modern item response theory (pp. 209-220). New York, NY: Springer-Verlag.

Sodhi, J., & Sodhi, P. (1999). Software reuse: Domain analysis and design process. New York, NY: McGraw-Hill.

Stefik, M. (1995). Introduction to knowledge systems. San Francisco, CA: Morgan Kaufmann.

Stone, R. B., Wood, K. L., & Crawford, R. H. (2000). A heuristic method for identifying modules for product architectures. Design Studies, 21, 5-31.

Taylor, W. (1953). CLOZE procedure: A new tool for measuring readability. Journalism Quarterly, 30, 414-438.

Tennyson, R. D. (1994). The big wrench vs. integrated approaches: The great media debate. Educational Technology Research & Development, 42 (3), 15-28.

Tennyson, R. D. & Park, O. (1984). Computer-based adaptive instructional systems: A review of empirically-based models. Machine-Mediated Learning, 1 (2), 129-149.

Tessmer, M., & Wedman, J. F. (1990). A layers-of-necessity instructional development model. Educational Technology Research & Development, 38 (2), 77-85.

Tripp, S. D., & Bichelmeyer, B. (1990). Rapid prototyping: An alternative instructional design strategy. Educational Technology Research & Development, 38 (1), 31-44.

Tutz, G. (1997). Sequential models for ordered responses. In W. J. van der Linden & R. K. Hambleton (Eds.), Handbook of modern item response theory (pp. 139-152). New York, NY: Springer-Verlag.

Ulrich, K. & Tung, K. (1991). Fundamentals of product modularity. Proceedings of the 1991 ASME Winter Annual Meeting Symposium on Issues in Design/Manufacturing Integration, Volume 39 Atlanta Georgia, 73-79.

WebCT (2000). WebCT software, version 3.0 Vancouver, BC: WebCT.

Weiss, D. J., & Kingsbury, G. G. (1984). Application of computerized adaptive testing to educational problems. Journal of Educational Measurement, 21 (4), 361-375.

Wieseler, W. (1999). RIO: A standards-based approach for reusable information objects. San Jose, CA: Cisco Systems, Inc. Available at: http://www.cisco.com

Wiley, D. A. (in press). The instructional use of learning objects. Bloomington, IN: Association for Educational Communications and Technology.

Wiley, D. A., South, J. B., Bassett, J., Nelson, L. M., Seawright, L., Peterson, T., & Monson, D. W. (1999). Three common properties of efficient online instructional support systems. Asynchronous Learning Networks Magazine, 3 (2). Available: http://www.aln.org/alnweb/magazine/Vol3_issue2/wiley.htm

Wilson, B. G., Jonassen, D. H., & Cole, P. (1993). Cognitive approaches to instructional design. In G. M. Piskurich (Ed.), The ASTD handbook of instructional technology (pp. 21.1-21.22). New York: McGraw-Hill.

Winne, P. H. (1993). A landscape of issues in evaluating adaptive learning systems. Journal of Artificial Intelligence in Education, 4 (4), 309-322.

Wu, A., & Tam, L. (1998). An architecture for dynamic courseware working on the Web. In H. Maurer & R. G. Olsen (Eds.), Proceedings of WebNet 98 – World Conference of the WWW, Internet & Intranet (pp.556-560). Charlottesville, VA: Association for the Advancement of Computing in Education.

Yourdon, E. (1994). Object-oriented system design: an integrated approach. Englewood Cliffs, NJ: Yourdon Press.

Appendix: Scripts for Common Item Types

Example 1: Multiple-Choice Item (w. Randomized Shuffling of Response Alternatives)

```
<questestinterop><item>
<itemrubric view="Candidate"><material><mattext>Select the single best answer.
</mattext></material></itemrubric>
<presentation>
<material><mattext>Which of the following is least affected by outliers?</mattext></material>
<response_lid ident="Outlier1"><render_choice shuffle="Yes">
<response_label ident="A"><material><mattext>Range</mattext></material></response_label>
<response_label ident="B"><material><mattext>Mean</mattext></material></response_label>
<response_label ident="C"><material><mattext>Mode</mattext></material></response_label>
<response_label ident="D"><material><mattext>Median</mattext></material></response_label>
</render_choice></response_lid>
</presentation>
<resprocessing>
<respcondition><conditionvar><varequal respident="Outlier1">C</varequal></conditionvar>
<setvar action="add" varname="score">1</setvar></respcondition></resprocessing>
</item></questestinterop>
```

Example 2: True-False Item (w. Predefined Order of Presentation of Response Alternatives)

```
<questestinterop>
<item>
<presentation>
<material><mattext>The mean is a measure of central tendency.</mattext></material>
<response_lid ident="Mean1"><render_choice shuffle="No">
<response_label ident="T"><material><mattext>True</mattext></material></response_label>
<response_label ident="F"><material><mattext>False</mattext></material></response_label>
</render_choice></response_lid>
```

```
</presentation>
<resprocessing>
<respcondition><conditionvar><varequal respident="Mean1">T</varequal></conditionvar>
<setvar action="add" varname="score">1</setvar></respcondition>
</resprocessing>
</item>
<item>
<presentation>
<material><mattext> Is the variance a measure of central tendency?</mattext></material>
<response_lid ident="Variance1"><render_choice shuffle="No">
<response_label ident="Y"><material><mattext>Yes</mattext></material></response_label>
<response_label ident="N"><material><mattext>No</mattext></material></response_label>
</render_choice></response_lid>
</presentation>
<resprocessing>
<respcondition><conditionvar><varequal respident="Variance1">N</varequal></conditionvar>
<setvar action="add" varname="score">1</setvar></respcondition>
</resprocessing>
</item>
</questestinterop>
```

Example 3: Rating Scale Item (Using Pointer to Common Material)

Note: the example below assumes that the following URIs are associated with the specified text strings: C1.txt = "Extremely"; C2.txt = "Quite a bit"; C3.txt = "Moderately"; C4.txt = "A little bit"; C5.txt = "Not at all".

```
<questestinterop><item>
```

```
<itemrubric view="Candidate"><material><mattext> For each of the following statements, select the response which best reflects the extent to which the problem bothered you in the last week.
</mattext></material></itemrubric>
<presentation>
<material><mattext> Thoughts of failing my statistics course.</mattext></material>
<response_lid ident="Failing"><render_choice shuffle="No">
<response_label ident="C1"><material><mattext uri="c1.txt"/></material></response_label>
<response_label ident="C2"><material><mattext uri="c2.txt"/></material></response_label>
<response_label ident="C3"><material><mattext uri="c3.txt"/></material></response_label>
<response_label ident="C4"><material><mattext uri="c4.txt"/></material></response_label>
<response_label ident="C5"><material><mattext uri="c5.txt"/></material></response_label>
</render_choice></response_lid>
<material><mattext>Feeling that I will never use statistical analysis in my professional life.</mattext></material>
<response_lid ident="Utility"><render_choice shuffle="No">
<response_label ident="C1"><material><mattext uri="c1.txt"/></material></response_label>
<response_label ident="C2"><material><mattext uri="c2.txt"/></material></response_label>
<response_label ident="C3"><material><mattext uri="c3.txt"/></material></response_label>
<response_label ident="C4"><material><mattext uri="c4.txt"/></material></response_label>
<response_label ident="C5"><material><mattext uri="c5.txt"/></material></response_label>
</render_choice></response_lid>
<material><mattext>Losing interest in studying statistics.</mattext></material>
<response_lid ident="Interest"><render_choice shuffle="No">
<response_label ident="C1"><material><mattext uri="c1.txt"/></material></response_label>
<response_label ident="C2"><material><mattext uri="c2.txt"/></material></response_label>
<response_label ident="C3"><material><mattext uri="c3.txt"/></material></response_label>
<response_label ident="C4"><material><mattext uri="c4.txt"/></material></response_label>
```

```
<response_label ident="C5"><material><mattext uri="c5.txt"/></material></response_label>
</render_choice></response_lid>
</presentation>
<resprocessing>
<respcondition><conditionvar><varequal respident="Failing">C1</varequal></conditionvar>
<setvar action=”add” varname="score">1</setvar></respcondition>
<respcondition><conditionvar><varequal respident="Failing">C2</varequal></conditionvar>
<setvar action=”add” varname="score">2</setvar></respcondition>
<respcondition><conditionvar><varequal respident="Failing">C3</varequal></conditionvar>
<setvar action=”add” varname="score">3</setvar></respcondition>
<respcondition><conditionvar><varequal respident="Failing">C4</varequal></conditionvar>
<setvar action=”add” varname="score">4</setvar></respcondition>
<respcondition><conditionvar><varequal respident="Failing">C5</varequal></conditionvar>
<setvar action=”add” varname="score">5</setvar></respcondition>
<respcondition><conditionvar><varequal respident="Utility">C1</varequal></conditionvar>
<setvar action=”add” varname="score">1</setvar></respcondition>
<respcondition><conditionvar><varequal respident="Utility">C2</varequal></conditionvar>
<setvar action=”add” varname="score">2</setvar></respcondition>
<respcondition><conditionvar><varequal respident="Utility">C3</varequal></conditionvar>
<setvar action=”add” varname="score">3</setvar></respcondition>
<respcondition><conditionvar><varequal respident="Utility">C4</varequal></conditionvar>
<setvar action=”add” varname="score">4</setvar></respcondition>
<respcondition><conditionvar><varequal respident="Utility">C5</varequal></conditionvar>
<setvar action=”add” varname="score">5</setvar></respcondition>
<respcondition><conditionvar><varequal respident="Interest">C1</varequal></conditionvar>
<setvar action=”add” varname="score">1</setvar></respcondition>
<respcondition><conditionvar><varequal respident="Interest">C2</varequal></conditionvar>
```

```
<setvar action="add" varname="score">2</setvar></respcondition>
<respcondition><conditionvar><varequal respident="Interest">C3</varequal></conditionvar>
<setvar action="add" varname="score">3</setvar></respcondition>
<respcondition><conditionvar><varequal respident="Interest">C4</varequal></conditionvar>
<setvar action="add" varname="score">4</setvar></respcondition>
<respcondition><conditionvar><varequal respident="Interest">C5</varequal></conditionvar>
<setvar action="add" varname="score">5</setvar></respcondition>
</resprocessing>
</item></questestinterop>
```

Example 4: Complex Multiple-Choice Item, Closed Set of Response Alternatives

```
<questestinterop><item>
<itemrubric view="Candidate"><material><mattext>Select the best answer.
</mattext></material></itemrubric>
<presentation>
<material><mattext>Which of the following descriptive statistics are measures of central
tendency?<mattext></material>
<material><mattext>1 = Mean<mattext></material>
<material><mattext>2 = Variance<mattext></material>
<material><mattext>3 = Mode<mattext></material>
<material><mattext>4 = Range<mattext></material>
<response_lid ident="Complex1"><render_choice>
<response_label ident="A"><material><mattext>1 and 3</mattext></material></response_label>
<response_label ident="B"><material><mattext>2 and 4</mattext></material></response_label>
<response_label ident="C"><material><mattext>1 only</mattext></material></response_label>
<response_label ident="D"><material><mattext>1, 2 and 3</mattext></material></response_label>
</render_choice></response_lid>
</presentation>
```

<resprocessing>

<respcondition><conditionvar><varequal respident="Complex1">A</varequal></conditionvar>

<setvar action="add" varname="score">3</setvar></respcondition>

<respcondition><conditionvar><varequal respident="Complex1">C</varequal></conditionvar>

<setvar action="add" varname="score">1</setvar></respcondition>

</resprocessing>

</item></questestinterop>

<u>Example 5: Complex Multiple-Choice Item, Open Set of Response Alternatives</u>

Note: the item below awards points within the following framework: 4 points for a correct response; 2 points for a response that includes no incorrect elements, but misses one of the correct elements; 1 points for a response that has all the correct elements for the response, but also includes one incorrect response; 0 points for all other responses.

<questestinterop><item>

<itemrubric view="Candidate"><material><mattext>For each answer option, indicate whether it is correct or incorrect.</mattext></material></itemrubric>

<presentation>

<material><mattext>Which of the following descriptive statistics are measures of central tendency?<mattext></material>

<response_lid ident="Mean"><material><mattext>Mean</mattext></material>

<render_choice>

<response_label ident="Y"><material><mattext>Measure of central tendency</mattext></material></response_label>

<response_label ident="N"><material><mattext>Not a measure of central tendency</mattext></material></response_label>

</render_choice></response_lid>

<response_lid ident="Variance">

<material><mattext>Variance</mattext></material>

<render_choice>
<response_label ident="Y"><material><mattext>Measure of central tendency</mattext></material></response_label>
<response_label ident="N"><material><mattext>Not a measure of central tendency</mattext></material></response_label>
</render_choice></response_lid>
<response_lid ident="Mode">
<material><mattext>Mode</mattext></material>
<render_choice>
<response_label ident="Y"><material><mattext>Measure of central tendency</mattext></material></response_label>
<response_label ident="N"><material><mattext>Not a measure of central tendency</mattext></material></response_label>
</render_choice></response_lid>
<response_lid ident="Range"><material><mattext>Range</mattext></material>
<render_choice>
<response_label ident="Y"><material><mattext>Measure of central tendency</mattext></material></response_label>
<response_label ident="N"><material><mattext>Not a measure of central tendency</mattext></material></response_label>
</render_choice></response_lid>
</presentation>
<resprocessing>
<respcondition continue=”no”>
<and>
<conditionvar><varequal respident="Mean">Y</varequal></conditionvar>
<conditionvar><varequal respident="Variance">N</varequal></conditionvar>

```
<conditionvar><varequal respident="Mode">Y</varequal></conditionvar>
<conditionvar><varequal respident="Range">N</varequal></conditionvar>
</and>
<setvar action=”add” varname="score">4</setvar>
</respcondition>
<respcondition continue=”no”>
<and>
<and>
<conditionvar><varequal respident="Range">N</varequal></conditionvar>
<conditionvar><varequal respident="Variance">N</varequal></conditionvar>
</and>
<or>
<conditionvar><varequal respident="Mode">Y</varequal></conditionvar>
<conditionvar><varequal respident="Mean">Y</varequal></conditionvar>
</or>
</and>
<setvar action=”add” varname="score">2</setvar>
</respcondition>
<respcondition continue=”no”>
<and>
<and>
<conditionvar><varequal respident="Mode">Y</varequal></conditionvar>
<conditionvar><varequal respident="Mean">Y</varequal></conditionvar>
</and>
<not><and>
<conditionvar><varequal respident="Range">Y</varequal></conditionvar>
<conditionvar><varequal respident="Variance">Y</varequal></conditionvar>
```

```
</and></not>
</and>
<setvar action="add" varname="score">1</setvar>
</respcondition>
</item></questestinterop>
```

Example 6: Fill-In-The-Blank Item

```
<questestinterop><item>
<presentation>
<response_str ident="CT1"><render_fib fibtype="string" prompt ="box" maxchars =
"20"><material><mattext>The mean is a measure of central </mattext></material><response_label
ident="placeholder"></response_label></render_fib></response_str>
</presentation>
<resprocessing>
<respcondition>
<or>
<conditionvar><varequal respident="CT1" case="no">tendency</varequal></conditionvar>
<conditionvar><varequal respident="CT1" case="no">tendancy</varequal></conditionvar>
<conditionvar><varequal respident="CT1" case="no">tendencies</varequal></conditionvar>
<conditionvar><varequal respident="CT1" case="no">tendancies</varequal></conditionvar>
</or>
<setvar action="add" varname="score">1</setvar>
</respcondition>
</resprocessing>
</item></questestinterop>
```

Example 7: CLOZE-Procedure Item

Note: the item below awards 1 point for each correct answer, with a bonus of 2 points if all three answers are correct.

```
<questestinterop><item>
<itemrubric view="Candidate"><material><mattext> In the paragraph below, fill in each blank with the missing word.</mattext></material></itemrubric>
<presentation>
<response_str ident="R1"><render_fib fibtype="string" prompt ="dashline" maxchars = "20">
<material><mattext>In place of the independence assumption, the Oneway Within-Groups ANOVA relies on the </mattext></material><response_label ident="placeholder"></response_label>
<material><mattext> assumption (also known as the sphericity assumption). </mattext></material>
</render_fib></response_str>
<response_str ident="R2"><render_fib fibtype="string" prompt ="dashline" maxchars = "20"><material><mattext>This assumption concerns the nature of the dependency among participants' scores across levels of the treatment (A) when there are more than </mattext></material>
<response_label ident="placeholder"></response_label><material><mattext> levels.
</mattext></material></render_fib></response_str>
<response_str ident="R3"><render_fib fibtype="string" prompt ="dashline" maxchars = "20">
<material><mattext> Except for chance fluctuation, one would expect the correlation among participant scores for any two levels of the treatment to be </mattext></material><response_label ident="placeholder"></response_label><material><mattext>. </mattext></material>
</render_fib></response_str>
</presentation>
<resprocessing>
<respcondition continue="yes"><conditionvar><varequal respident="R1" case="no">circularity</varequal></conditionvar><setvar action="add" varname="score">1</setvar></respcondition>
<respcondition continue="yes"><conditionvar><varequal respident="R2" case="no">two</varequal></conditionvar><setvar action="add" varname="score">1</setvar></respcondition>
```

```
<respcondition continue="yes"><conditionvar><varequal respident="R3"
case="no">equal</varequal></conditionvar><setvar action="add"
varname="score">1</setvar></respcondition>
<respcondition>
<and>
<conditionvar><varequal respident="R1" case="no">circularity</varequal></conditionvar>
<conditionvar><varequal respident="R2" case="no">two</varequal></conditionvar>
<conditionvar><varequal respident="R3" case="no">equal</varequal></conditionvar>
</and>
<setvar action="add" varname="score">2</setvar></respcondition>
</resprocessing>
</item></questestinterop>
```

Example 8: Essay Item

```
<questestinterop><item>
<presentation>
<material><mattext> List and describe the threats to internal validity in a one-shot case study.
</mattext></material>
<response_str ident="R1"><render_fib fibtype="string" prompt ="box" rows = "100" columns="80">
<response_label ident="placeholder"></response_label></render_fib></response_str>
</presentation>
<resprocessing></resprocessing>
</item></questestinterop>
```

Example 9: Numerical Item, Specification of Response Events as Points

```
<questestinterop><item>
<itemrubric view="Candidate"><material><mattext>For the following item, round your response to two
decimal places.</mattext></material></itemrubric>
```

```
<presentation><material><mattext>The following data were gathered during an experiment:
</mattext></material>
<material><mattext>Treatment group: 11, 9, 12, 9, -2, 7, 7, 16, 7, 10, 2, 13, 15</mattext></material>
<material><mattext>Control group: 9, 7, 7, 4, 8, 8, 7, 6, 3, 11, 6, 8, 5 </mattext></material>
<response_num ident="R1">
<material><mattext> What is the mean of the treatment group?</mattext></material>
<render_fib fibtype=”decimal” prompt =”dashline” maxchars=”6”><response_label
ident=”placeholder”></response_label></render_fib>
</response_num>
</presentation>
<resprocessing>
<respcondition><conditionvar><varequal respident="R1">8.92</varequal></conditionvar>
<setvar action=”add” varname="score">1</setvar></respcondition>
</resprocessing>
</item></questestinterop>
```

Example 10: Numerical Item, Specification of Response Events as Ranges

```
<questestinterop><item>
<itemrubric view="Candidate"><material><mattext>For the following item, round your response to two
decimal places.</mattext></material></itemrubric>
<presentation>
<material><mattext>The following data were gathered during an experiment: </mattext></material>
<material><mattext>Treatment group: 11, 9, 12, 9, -2, 7, 7, 16, 7, 10, 2, 13, 15</mattext></material>
<material><mattext>Control group: 9, 7, 7, 4, 8, 8, 7, 6, 3, 11, 6, 8, 5 </mattext></material>
<response_num ident="R1">
<material><mattext> What is the mean of the treatment group?</mattext></material>
<render_fib fibtype=”decimal” prompt =”dashline” maxchars=”6”><response_label
ident=”placeholder”></response_label></render_fib></response_num>
```

```
</presentation>
<resprocessing>
<respcondition>
<and>
<conditionvar><vargte respident="R1">8.9</varequal></conditionvar>
<conditionvar><varlt respident="R1">9.0</varequal></conditionvar>
</and>
<setvar action="add" varname="score">1</setvar>
</respcondition>
</resprocessing>
</item></questestinterop>
```

<u>Example 11: Item Set</u>

```
<questestinterop><item>
<presentation>
<material><mattext>The following data were gathered during an experiment: </mattext></material>
<material><mattext>Treatment group: 11, 9, 12, 9, -2, 7, 7, 16, 7, 10, 2, 13, 15</mattext></material>
<material><mattext>Control group: 9, 7, 7, 4, 8, 8, 7, 6, 3, 11, 6, 8, 5 </mattext></material>
<response_lid ident="R1"><material><mattext> What is the mean of the treatment
group?</mattext></material>
<render_choice>
<response_label ident="A"> <material><mattext>8.92</mattext></material></response_label>
<response_label ident="B"> <material><mattext>9.07</mattext></material></response_label>
</render_choice></response_lid>
<response_lid ident="R2">
<material><mattext> What is the mean of the control group?</mattext></material>
<render_choice>
<response_label ident="A"> <material><mattext>6.85</mattext></material></response_label>
```

```
<response_label ident="B"> <material><mattext>7.42</mattext></material></response_label>
</render_choice></response_lid>
</presentation>
<resprocessing>
<respcondition continue="no">
<and>
<conditionvar><varequal respident="R1">A</varequal></conditionvar>
<conditionvar><varequal respident="R2">A</varequal></conditionvar>
</and>
<setvar action="add" varname="score">3</setvar></respcondition>
<respcondition continue="yes"><conditionvar><varequal respident="R1">A</varequal></conditionvar>
<setvar action="add" varname="score">1</setvar></respcondition>
<respcondition><conditionvar><varequal respident="R2">A</varequal></conditionvar>
<setvar action="add" varname="score">1</setvar></respcondition>
</resprocessing>
</item></questestinterop>
```

<u>Example 12: True-False Item with Corrections</u>

```
<questestinterop><item>
<itemrubric view="Candidate"><material><mattext>For the following question, select either TRUE or
FALSE. If you select FALSE, enter the correct term in the space provided.
</mattext></material></itemrubric>
<presentation>
<material><mattext>The mean is a measure of dispersion.</mattext></material>
<response_lid ident="R1"><render_choice>
<response_label ident="T"><material><mattext>TRUE</mattext></material></response_label>
<response_label ident="F"><material><mattext>FALSE</mattext></material></response_label>
</render_choice></response_lid>
```

```
<response_str ident="R2"><material><mattext>The mean is a measure of </mattext></material>
<render_fib  fibtype=”string” prompt =”dashline” maxchars = “20”>
<response_label ident=”placeholder”></response_label>
</render_fib></response_str>
</presentation>
<resprocessing>
<respcondition continue=”no”>
<and>
<conditionvar><varequal respident="R1">F</varequal></conditionvar>
<conditionvar><varequal respident="R2" case=”no”>central tendency</varequal></conditionvar>
</and>
<setvar action=”add” varname="score">3</setvar></respcondition>
<respcondition continue=”no”>
<conditionvar><varequal respident="R1">F</varequal></conditionvar>
<setvar action=”add” varname="score">1</setvar></respcondition>
</resprocessing>
</item></questestinterop>
```

Example 13: Problem-Solving Item Set (w. Diagnostic Response Processing)

Note: This item awards 5 points if all four questions are answered correctly. 1 point is given for each correct answer. 1 point is also given for an incorrect estimated population variance if the sole reason why it is incorrect is that an incorrect mean was used from a previous response.

```
<questestinterop><item>
<presentation>
<material><mattext>The following data were gathered during an experiment:</mattext></material>
<material><mattext>Treatment group: 7, 5, 6, 5, 7</mattext></material>
<material><mattext>Control group: 3, 4, 2, 5, 2</mattext></material>
```

<response_lid ident="R1"><material><mattext> What is the mean of the treatment
group?</mattext></material>
<render_choice shuffle="Yes"><response_label ident=”TreatmentMean_Correct”>
<material><mattext>6.0</mattext></material></response_label>
<response_label ident=”TreatmentMean_Incorrect_DivideByN-1”>
<material><mattext>7.5</mattext></material></response_label></render_choice></response_lid>
<response_lid ident="R2"><material><mattext> What is the mean of the control
group?</mattext></material><render_choice shuffle="Yes">
<response_label ident=”ControlMean_Correct”>
<material><mattext>3.2</mattext></material></response_label>
<response_label ident=”ControlMean_Incorrect_DivideByN-1”>
<material><mattext>4.0</mattext></material></response_label></render_choice></response_lid>
<response_lid ident="R3"><material><mattext>What is the estimated population variance of the treatment
group?</mattext></material><render_choice shuffle="Yes">
<response_label ident=”TreatmentPopVar_Correct”>
<material><mattext>0.80</mattext></material></response_label>
<response_label ident=”TreatmentPopVar_Incorrect_DivideByN”>
<material><mattext>1.00</mattext></material></response_label>
<response_label ident=”TreatmentPopVar_Incorrect_WrongMean”>
<material><mattext>2.85</mattext></material></response_label>
<response_label ident=”TreatmentPopVar_Incorrect_WrongMeanAndDivideByN”>
<material><mattext>3.56</mattext></material></response_label>
</render_choice></response_lid>
<response_lid ident="R4">
<material><mattext>What is the estimated population variance of the control group?</mattext></material>
<render_choice shuffle="Yes"><response_label ident=”ControlPopVar_Correct”>
<material><mattext>1.39</mattext></material></response_label>

```
<response_label ident=”ControlPopVar_Incorrect_DivideByN”>
<material><mattext>1.74</mattext></material></response_label>
<response_label ident=”ControlPopVar_Incorrect_IncorrectMean”>
<material><mattext>2.00</mattext></material></response_label>
<response_label ident=”ControlPopVar_Incorrect_IncorrectMeanAndDivideByN”>
<material><mattext>2.50</mattext></material></response_label>
</render_choice></response_lid>
</presentation>
<resprocessing>
<respcondition continue=”no”>
<and>
<conditionvar><varequal respident="R1">TreatmentMean_Correct</varequal></conditionvar>
<conditionvar><varequal respident="R2">ControlMean_Correct</varequal></conditionvar>
<conditionvar><varequal respident="R3">TreatmentPopVar_Correct</varequal></conditionvar>
<conditionvar><varequal respident="R4">ControlPopVar_Correct</varequal></conditionvar>
</and>
<setvar action=”add” varname="score">5</setvar></respcondition>
<respcondition continue=”yes”>
<conditionvar><varequal respident="R1">TreatmentMean_Correct</varequal></conditionvar>
<setvar action=”add” varname="score">1</setvar></respcondition>
<respcondition continue=”yes”>
<conditionvar><varequal respident="R2">ControlMean_Correct</varequal></conditionvar>
<setvar action=”add” varname="score">1</setvar></respcondition>
<respcondition continue=”yes”>
<or>
<conditionvar><varequal respident="R3">TreatmentPopVar_Correct</varequal></conditionvar>
<and>
```

```
<conditionvar><varequal
respident="R3">TreatmentPopVar_Incorrect_IncorrectMean</varequal></conditionvar>
<conditionvar><varequal respident="R1">TreatmentMean_Incorrect</varequal></conditionvar>
</and></or>
<setvar action=”add” varname="score">1</setvar></respcondition>
<respcondition>
<or>
<conditionvar><varequal respident="R4">ControlPopVar_Correct</varequal></conditionvar>
<and>
<conditionvar><varequal
respident="R4">ControlPopVar_Incorrect_IncorrectMean</varequal></conditionvar>
<conditionvar><varequal respident="R2">ControlMean_Incorrect</varequal></conditionvar>
</and></or>
<setvar action=”add” varname="score">1</setvar></respcondition>
</resprocessing>
</item></questestinterop>
```

<u>Example 14: Matching Item</u>

```
<questestinterop><item>
<itemrubric view="Candidate"><material><mattext>Match the nonparametric tests with their parametric
equivalents. Enter the number of each parametric test beside the nonparametric test it is equivalent
to.</mattext></material></itemrubric>
<presentation>
<material><mattext>(1). Student's t-test</mattext></material>
<material><mattext>(2). Correlated samples t-test</mattext></material>
<material><mattext>(3). Within-Groups ANOVA</mattext></material>
<material><mattext>(4). Oneway ANOVA</mattext></material>
<response_num ident="R1"><render_fib fibtype=”integer” prompt =”dashline” maxchars=”1”>
```

```
<response_label ident=”placeholder”></response_label>
<material><mattext>Friedman ANOVA.</mattext></material></render_fib></response_num>
<response_num ident="R2"><render_fib fibtype=”integer” prompt =”dashline” maxchars=”1”>
<response_label ident=”placeholder”></response_label>
<material><mattext>Kruskal-Wallis H test.</mattext></material></render_fib></response_num>
<response_num ident="R3"><render_fib fibtype=”integer” prompt =”dashline” maxchars=”1”>
<response_label ident=”placeholder”></response_label>
<material><mattext>Wilcoxon signed-ranks t test.</mattext></material>
</render_fib></response_num>
<response_num ident="R4"><render_fib fibtype=”integer” prompt =”dashline” maxchars=”1”>
<response_label ident=”placeholder”></response_label>
<material><mattext>Mann-Whitney U test.</mattext></material>
</render_fib></response_num>
</presentation>
<resprocessing>
<respcondition continue=”yes”><conditionvar><varequal respident="R1">3</varequal></conditionvar>
<setvar action=”add” varname="score">1</setvar></respcondition>
<respcondition continue=”yes”><conditionvar><varequal respident="R2">4</varequal></conditionvar>
<setvar action=”add” varname="score">1</setvar></respcondition>
<respcondition continue=”yes”><conditionvar><varequal respident="R3">2</varequal></conditionvar>
<setvar action=”add” varname="score">1</setvar></respcondition>
<respcondition continue=”yes”><conditionvar><varequal respident="R4">1</varequal></conditionvar>
<setvar action=”add” varname="score">1</setvar></respcondition>
</resprocessing>
</item></questestinterop>
```

Example 15: Ordering Item

```
<questestinterop><item>
```

<itemrubric view="Candidate"><material><mattext>Place the following steps in hypothesis testing into the correct order. Write 1 beside the first step, 2 beside the second step, and so on.</mattext></material></itemrubric>
<presentation>
<response_num ident="R1"><render_fib fibtype="integer" prompt ="dashline" maxchars="1"><response_label ident="placeholder"></response_label><material><mattext>Compute the calculated value of the test statistic.</mattext></material></render_fib></response_num>
<response_num ident="R2"><render_fib fibtype="integer" prompt ="dashline" maxchars="1"><response_label ident="placeholder"></response_label><material><mattext>State the null and alternative hypotheses.</mattext></material></render_fib></response_num>
<response_num ident="R3"><render_fib fibtype="integer" prompt ="dashline" maxchars="1"><response_label ident="placeholder"></response_label><material><mattext>Find the critical value of the test statistic.</mattext></material></render_fib></response_num>
<response_num ident="R4"><render_fib fibtype="integer" prompt ="dashline" maxchars="1"><response_label ident="placeholder"></response_label><material><mattext>Select the appropriate test statistic.</mattext></material></render_fib></response_num>
<response_num ident="R5"><render_fib fibtype="integer" prompt ="dashline" maxchars="1"><response_label ident="placeholder"></response_label><material><mattext>Compare the calculated and critical values.</mattext></material></render_fib></response_num>
<response_num ident="R6"><render_fib fibtype="integer" prompt ="dashline" maxchars="1"><response_label ident="placeholder"></response_label><material><mattext>Select alpha.</mattext></material></render_fib></response_num>
<response_num ident="R7"><render_fib fibtype="integer" prompt ="dashline" maxchars="1"><response_label ident="placeholder"></response_label><material><mattext>Make a decision about the null and alternative hypotheses.</mattext></material></render_fib></response_num>
</presentation>
<resprocessing>

```
<respcondition continue=”yes”><conditionvar><varequal respident="R1">4</varequal></conditionvar>
<setvar action=”add” varname="score">1</setvar></respcondition>
<respcondition continue=”yes”><conditionvar><varequal respident="R2">1</varequal></conditionvar>
<setvar action=”add” varname="score">1</setvar></respcondition>
<respcondition continue=”yes”><conditionvar><varequal respident="R3">5</varequal></conditionvar>
<setvar action=”add” varname="score">1</setvar></respcondition>
<respcondition continue=”yes”><conditionvar><varequal respident="R4">3</varequal></conditionvar>
<setvar action=”add” varname="score">1</setvar></respcondition>
<respcondition continue=”yes”><conditionvar><varequal respident="R5">6</varequal></conditionvar>
<setvar action=”add” varname="score">1</setvar></respcondition>
<respcondition continue=”yes”><conditionvar><varequal respident="R6">2</varequal></conditionvar>
<setvar action=”add” varname="score">1</setvar></respcondition>
<respcondition continue=”yes”><conditionvar><varequal respident="R7">7</varequal></conditionvar>
<setvar action=”add” varname="score">1</setvar></respcondition>
</resprocessing>
</item></questestinterop>
```

Example 16: Simulation Item (Using an External Application)

```
<questestinterop><item>
<itemrubric view="Candidate"><material><mattext>Using the circuit simulator, create a series electrical
circuit with a total resistance of 10 ohms.</mattext></material></itemrubric>
<presentation>
<material><matapplication uri=”http://www.somewhere.com/circuit.dcr”></matapplication></material>
<response_str ident="CircuitType"><render_extension>
<circuit_type><response_label=”placeholder”></response_label></circuit_type>
</render_extension></response_num>
<response_num ident="Resistance"><render_extension>
<circuit_totalresistance><response_label=”placeholder”></response_label></circuit_type>
```

```
</render_extension></response_num></presentation>
<resprocessing><respcondition>
<and>
<conditionvar><varequal respident="CircuitType">series</varequal></conditionvar>
<conditionvar><vargte respident="Resistance">9.5</varequal></conditionvar>
<conditionvar><varlte respident="Resistance">10.5</varequal></conditionvar>
</and>
<setvar action="add" varname="score">3</setvar></respcondition></resprocessing>
</item></questestinterop>
```