IMPART: A Prototype Authoring System for Interactive Multimedia Vocabulary Tutorials and Drills

Brockenbrough S. Alien
Steven L. Eckols

Abstract: This article describes IMPART, a prototype authoring system for foreign language vocabulary training. Developed with Apple Computer's HyperCard, the authoring system lets developers create sophisticated drill and practice lessons using a range of representational modalities including text, graphics, "natural video," and digitized audio. The authors describe the goals of the project, the components of the authoring system, and the design rationale for the system's major features.

IMPAKT is a prototype authoring system for developing computer-based interactive multimedia (Ambron & Hooper, 1988) drill and practice lessons. It was designed to support vocabulary drills in any foreign language that can be represented by Macintosh keyboard characters and was developed as a tool for foreign language professors and instructors at San Diego State University's Language Acquisition Resource Center. Lesson authors can specify the items presented in a lesson as well as underlying assumptions about drill operations. The system can therefore also be used as a tool for research on drill mechanics. However, it is primarily intended to aid authors in creating lessons that teach associations between pairs of objects (symbol-word, picture-word, word-word, etc.) when these require a degree of rote memorization. Although it is intended primarily as an adjunct to foreign language instruction, IMPART can accommodate content from other disciplines and could be used to teach paired associate learning tasks in content domains ranging from mathematics to biology.

Paired Associate Learning Tasks

The fundamental problem in supporting paired associate learning (Bower & Hilgard, 1981) is to facilitate acquisition in the learner's mind of memory links or associations between pairs of words or other stimuli. Performance criteria for this type of learning task require that the learner master a set of paired elements and respond with one element of each pair when presented with the other element. Paired associateshave been used widely by experimental psychologists to study a variety of memory effects.

The ability to respond to a stimulus element with its appropriate associate is influenced by numerous factors, such as the total number of pairs in the set, the similarity of elements, and whether the pairings are based on "meaningful" associations. From the learner's point of view, the pairing of elements in a set usually seems arbitrary at first. Indeed, the appropriateness of the techniques enabled by IMPART is mostly restricted to learning problems that involve "fact" or "rote" content (i.e., content in which associations are essentially historical in nature or derived from systems of meaning unknown to the learner). From the standpoint of lesson design, the most important problems are to help learners to establish a solid "link" or association between each of the paired elements in the set and to manage the process of linking in a way that minimizes the confusion of links among the various pairs in the set.

Language theorists and teachers disagree about the value of paired associate learning as an adjunct to language instruction. The system designers were heavily influenced by the knowledge that a successful authoring system for teaching foreign language vocabulary must be flexible and able to accommodate a variety of approaches and methods.

OVERVIEW OF THE SYSTEM

IMPART helps lesson designers to select and manage: a) the representational modality (text, speech, picture) of the stimulus (responses are always represented as text); b) the context or meaning of the link between the paired elements; and c) the events involved in rehearsal of stimulus-response pairs. As currently configured, the system is designed to work with a Macintosh SE or Macintosh II connected to a variety of standard videodisc players through the Macintosh phone port. (Use of the videodisc player is optional.) Performance on the Macintosh Plus is slow.

The system provides two execution environments: one for lesson development and one for lesson delivery. The first is called the author stack, and the second is called the student stack. ("Stack" is the term used for a HyperCard "program" file.) Data entered through the author stack is transferred to the student stack via an intermediate ASCII text file. Since lesson data is stored in a separate file, the student stack is an independent, general tool; it can work with any number of different lesson files. The ASCII lesson files could also be used by drill and practice programs written in other computer languages and for other delivery platforms. Following sections describe the features of the author and student stacks in more detail.

The Development Environment

The author stack provides a data-^ntry environment in which a lesson developer specifies the information required by the student stack. The author first determines certain general features of the lesson that will apply throughout the learner's interactions, including the number of options that will be provided for multiple-choice questions, the size of various item pools involved in the management of the lesson, and the criteria for defining mastered items. Then the author enters information for each of the vocabulary items that will comprise the lesson. The minimum element set for each item is two: a character string representing the word or phase in the native language, and a string representing the word in the target language. Although specifying strings for the target and native language equivalents for each item is all that is required for a fully functional lesson, use of the system's multimedia capabilities requires additional information for some or all of the items. The author can specify videodisc scenes in which the word is spoken in context, a bit-mapped graphic representing the word, isolated digital audio pronunciation of the word, and various types of supplementary information such as cognates and grammatical notes.

The author can also specify the timing and conditions under which each of these additional representations will be made available to the learner. For example, the author can specify that a particular representation will be available at any time under learner control. On the other hand, the author can specify that the representation appear automatically as feedback after the student has failed the item during practice.

The Delivery Environment

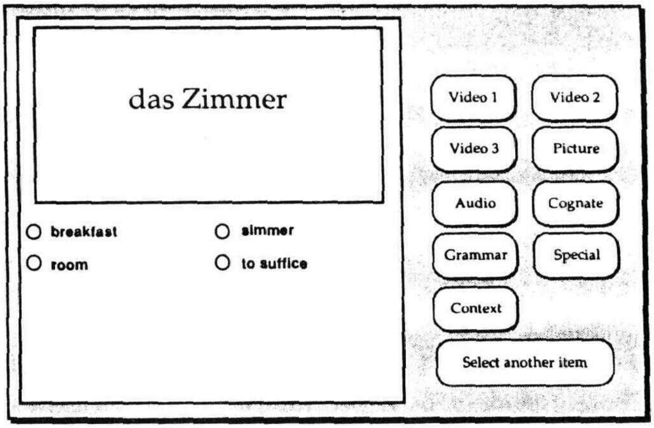
The features of the authoring stack are most easily understood from the point of view of the learner, so they are described in this section, which presents the system's delivery environment, the student stack. It has four components: a video preview shell, a tutorial shell, a drill shell, and a quiz shell. Although the sequence in which the learner encounters these components (and indeed, whether they are encountered at all) is under learner control, the system is designed to cue the learner to experience them in the sequence listed above.

The video preview shell. This is the first component of the student stack. The system allows the author to specify a videodisc segment from which all (or many) of the items in the lesson were drawn. If a videodisc player is available, the learner may play the entire segment.

The tutorial shell. The second component of the student stack displays a scrolling list of all of the items in the lesson. The learner may select any item for further study. The student stack then formats the item as a practice as shown in Figure 1 (see the following page).

In this case, the selected item, "das Zimmer" (German for "room") is accompanied by various learner-control options specified by the author. The target language word is presented along with four multiple-choice responses. (This is the same format that is used in the drill and quiz components of the student stack, which the learner will encounter after leaving the tutorial component.) The number of options for the multiple-choice format is determined by the author and can range from two to ten. As an alternative, the author can also

Figure 1.
A Screen from the Tutorial Component of the Student Stack.



specify a constructed response format (fill-in the space). However, the response must be correctly spelled. The system does not use "relaxed" spelling options when it evaluates a response. Since IMPART was to be used for multilingual applications, this was judged too difficult to implement.

The initial stimuli for the practice items need notbe restricted to character strings in the target language. The author can simply specify that native language words be used as the stimuli for all items in a lesson, in which case multiple-choice options will be automatically displayed in the target language. The author can also specify that a bit-mapped graphic, video segment, or digital "sound bite" be used as the stimulus for all items in a lesson.

The student stack uses native language strings entered by the author for other items as options for each target-language item. This convenient IMPART feature makes it unnecessary for the author to design specific items for practices or quizzes; they are constructed automatically. Furthermore, since the student stack uses pseudo-random routines each time it generates the options for a multiple-choice question, the configuration of options will be different each time the student encounters "das Zimmer."

During lesson development, an author can designate a specific "foil" (plausible distractor) for each item. A foil is an incorrect answer that the student stack frequently makes available as an option for a particular vocabulary item. For instance, in Figure 1, the option "simmer" is a foil for "das Zimmer." A student who does not know the equivalent for "das Zimmer" might select "simmer" simply because it is phonetically similar and might therefore be assumed to be a cognate. When a foil is specified by the author for a particular item, it is used in multiple-choice questions not only in the tutorial component of the student stack, but also in the drill and quiz components.

The author can also designate a fell for use when items will be formatted with the *native* language string as the stimulus. In such cases, the options will be selected from among the *target* language equivalents for other words in the lesson. The foil designated by the author could be *semantically* similar to the native language string and may be chosen to sharpen the student's discrimination of related concepts. "Platz" (space), "Saal" (hall), or "Wohnung" (dwelling) might be used in this way for "das Zimmer."

The tutorial is designed to provide a variety of information elements and representational modalities for a single item. As can be seen in Figure 1, the learner has the option of "clicking" on HyperCard buttons to:

a) see one of three videodisc segments which incorporate the phrase "das Zimmer" in conversations;
b) view a computer graphic (picture) representing the phrase;
c) hear a digitized sound recording of the item spoken by a native speaker;
d) see a cognate of the item;
e) see grammatical details about the item;
f) examine special information about the item, such as phonetic spelling; and
g) see a sample sentence with the item used in context.

The specific circumstances in which any of the informational elements and representational modalities are made available to the learner is also designated by the lesson author. For instance, the author may make the "Video 1" option available before the student makes any response for an item, but may specify that the "Video 2" option appear only after *the first* incorrect response. Further, the author may specify that the "Video 3" option be made available only after the student has responded incorrectly twice. It should be noted that even if most or all of the range of representations is available to the learner, he or she is not forced to select any of them.

The drill shell. Unlike the tutorial, the drill is not controlled by the student. Instead, it forces the student through a "lock-step" experience designed to promote rapid responses with little opportunity for reflection. The specific items and options presented, the sequence of these items, and the number of times they are presented are all determined automatically by the student stack according to parameters set by the lesson author.

Routines for managing the drill (see Figure 2) employ a modified version of the designs described by Salisbury (1988). IMPART actually employs four pools: lesson, working, review, and mastery. Items migrate through these pools according to parameters set by the author.

a) At the beginning of a drill session, all of the items in the lesson are in the first pool, the *lesson pool.*
b) When the student initiates the drill, the student stack selects enough items to fill the second pool, the *working pool,* the size of which is determined by the author.

c) Items are drawn at random from the working pool and presented to the student. If the lesson author has specified that the multiple-choice format be used, and if a "foil" string has been entered for an item) the foil will appear from time to time as one of the multiple-choice options.
d) Feedback for student responses is nearly instantaneous (limited only by the speed of the software and the computer), direct (right or wrong), and unaccompanied by additional information.
e) When a student responds correctly to an item for a specified number of times (set by the author), the item is promoted to the *review pool.*
f) The student stack fills the newly available slot in the working pool by selecting a new item at random, alternating between the lesson pool and the review pool.
g) When an item has migrated through the review pool-working pool loop a number of times set by the author, it is moved into the mastery pool.
h) When all of the items have migrated to the mastery pool, the drill is over.

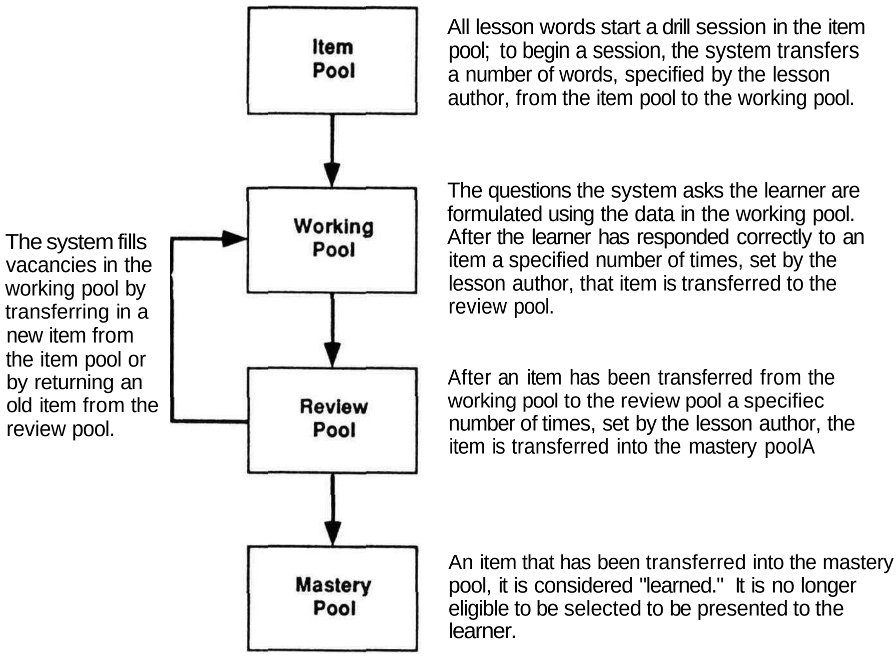
A drill session can last a long time if a student consistently gives incorrect answers, if the pool transfer criteria are rigorous, or if a lesson contains a large number of items.

The quiz shell. The fourth and last component of the student stack is the quiz shell. The quiz is to be taken by the student after he or she has completed the drill. It presents each of the items in the lesson in the same format as the drill shell. However, each item is presented once and only once and without feedback. After the quiz shell has presented all of the items in the lesson, it displays the student's score (as a percentage). The system can also display the student's answers accompanied by correct answers.

Observations

IMPARTs authoring stack accommodates a wide range of involvement on the part of the lesson author. At one extreme, the author might simply select a list of word pairs and have them entered by a clerical assistant. If the author considers the default settings for the student stack (target language string with four-option multiple-choice items) to be acceptable, nothing more need be done; the lesson file will run on the student stack without further specifications. At the other extreme, the author might elect to develop complex lessons with numerous adjunct representations and advisories, answers with various conditioned representations, and informational elements (e.g., display after second incorrect response).

Figure 2.
Pool Structure of the Drill Component of the Student Stack.



RATIONALE FOR THE SYSTEM'S FEATURES

We attempted to apply sound principles of instructional design throughout the development of the IMPART system. However, some of these principles were applied consciously while others were applied unconsciously. Some design decisions and ideas were justified at the time they were made through explicit reference to published models, theories, and empirical research. In some cases, we deliberately accommodated competing theories or models in the IMPART prototype. Other design decisions, both conscious and unconscious, were the result of "hunches" not explicitly supported by formal theories, principles, or empirical evidence. Additional features were included in response to practical concerns and exigencies of the moment or in anticipation of potential lesson author preferences.

Use of general (and partially unconscious) knowledge permits developers to work at a more creative, higher level, free from the constraints of particular models but at the same time subtly guided by them all. This approach had a positive impact on the project. However, in addition to the holistic effects of broad principles and general experience, there were specific research-based principles that directly influenced the design of IMPART.

In the following sections, we attempt to distinguish between prospective justification and retrospective justification of various IMPART features. *Prospective* justifications are those which we recall having made (in part at least) *prior to* or concurrently with the design decision to which they apply. *Retrospective* justifications are those which we recall having made subsequent to the design decision. Most of our retrospective justifications are based on literature reviews conducted after major development work was completed.

It may appear self-serving to offer retrospective justification for IMPART's features, but it is not unrealistic. After all, how many development projects are really planned from the start based explicitly and exclusively on a specific model or theory? In our opinion, many (if not most) instructional development decisions are based on the general training and experience of the developers. Hopefully this includes exposure to a range of theories and principles.

Why IMPART is Structured As It Is

The four shells that constitute the student stack were not directly derived from any formal design model, but in retrospect they reflect Gagne's (1970) "Nine Events of Instruction." (Several of the events are omitted however; Gagne"'s model does not require inclusion of all nine events.) The video preview serves to engage the learner, gain his or her attention (Event 1) and remind him or her of the items to be presented in the lesson (Event 3). Key vocabulary items may have already been presented in another instructional setting. To the extent that they have been, the video preview may also stimulate recall of prerequisite learning (Event 3). The tutorial shell presents the stimulus material (Event 4). The drill shell elicits performance (Event 6) and provides feedback about performance (Event 7). Finally, the quiz shell assesses performance (Event 8).

Why IMPART Provides for a Range of Learner Control Options

Every CAI designer must confront the issue of learner control. Although we were familiar with the general literature on learner control as it has evolved over the last 10 years, our deliberations were not derived directly from specific literature-based prescriptions. What follows is therefore in the nature of a retrospective justification.

Bonner (1988) argues that instructional products are often overly prescriptive. Brown (1986) claims that learner control over instruction is motivating. On the other side of the issue, Jonassen (1986, p. 287) cites a series of studies that show that "the case for learner control of instruction, which requires self-determination, autonomy, and responsibility, simply has not been empirically supported."

Hannafin (1984) proposes a continuum in which, at one extreme ("learner control"), sequence is determined completely by the learner; at the other extreme ("lesson control"), sequence is determined solely by the delivery system. He argues that selection of a location on the continuum should be determined by the characteristics of both learners and the material to be presented. Required learning outcomes are obviously another important consideration.

IMPART provides for degrees of learner control all along the Hannafin continuum. However, the author stack permits authors to specify many learner control options in the tutorial shell of the student stack; learner control is drastically constrained in the other components of the student stack.

The decision to select any one of the four basic components (preview, tutorial, drill, quiz) is, of course, left to the student. IMPART allows lesson authors to adjust the amount of learner control for each item in the tutorial. The author can vary the number of available representations from one element (presentation of only one stimulus - usually the word string) to several representations which may or may not be eventually selected for exploration by the student. The drill shell sacrifices learner control to the need for automaticity training, but it does adapt the presentation of items to the student's response patterns. The quiz shell, of course, totally eliminates learner control.

Why IMPART Uses a Pool Structure for the Drill

Whenever technology is considered as a means for addressing learning problems, designers should consider whether non-technological solutions might be just as effective. In the case of computer-based drills, the title of David Salisbury's article "When is a Computer Better than Flashcards?" (1988) is right to the point. It directly influenced the design of IMPART as did other work by Salisbury cited in this article.

Klein and Salisbury (1987) have demonstrated that flashcards can produce learning results that match those achieved through sophisticated computer-based drills. However, Klein and Salisbury note that the learners in their study demonstrated well developed learning strategies and suggest that learners with less well developed strategies can benefit from computer-based drills. The four-pool drill structure in the IMPART system was planned with this in mind.

Edwards and Siegal (1985), argue that simple drill and practice programs are flawed at two extremes. If the number of items is large, the learner is likely to forget missed items before he or she has a chance to answer them again. If the number of items presented is small, the learner will not be required to remember a missed item for any extended period, and long-term retention will suffer.

The four-pool structure used to manage the IMPART drill component addresses both of these issues. Items that are presented to the learner are drawn from a relatively small working pool, regardless of the total number of items that make up the lesson. However, the size of the working pool (which is specified by the author) need not be so small that an item will be fresh in the learner's mind when it is next encountered. The use of the review pool from which working pool vacancies are filled ensures that "learned" items are intermittently re-presented to the student to verify that they really were 'learned" and to promote long-term retention.

An important goal offoreign language vocabulary instruction is to promote automaticity in processing the meaning of words. Anderson (1980) describes automaticity as the state in which a practiced process requires little, if any, attention. Automaticity is an important factor in language learning because one cannot be fluent in a second language if much attention has to be focused on remembering commonly used words. According to Salisbury, Richards, and Klein (1985), effective automaticity training has three stages: a) accurate practice, b) accurate and fast practice, and c) accurate, fast, and "burdened" practice, in which the learner must divide attention between the drill exercise and some competing activity. The drill component ofthe student stack provides an environment in which learners can experience the first two of these three stages.

Why IMPART Provides for a Wide Range ofRepresentational Modalities

The wide range ofrepresentational modalities available in IMPART might seem to run counter to views held by authors such as Clark (1983) who argue that the various "media" used to deliver instructional messages have no differential impact on learning outcomes. Clark's review ofthe literature offers a number ofrival hypotheses to explain studies over the last two decades that purport to demonstrate the superiority ofone "medium" over others as vehicles for delivering instruction. In our view, however, Clark's arguments are based on a definition of media as hardware/delivery systems. It was "media" in the sense of communications modalities - basic systems that humans use for encoding information - such as speech, text, and pictorial codes that we wanted to exploit.

From the point of view of the lesson author, the choice of modalities depends partly on the intended learning outcomes. Should the criterion be based on the ability to spell words - as might be the case if instruction is to support acquisition of writing skills? Or is assessment of mastery to be based on ability to select a response to a text stimulus - as might be the case if instruction is to support reading? Stimuli consisting of spoken words might be used to support instruction in comprehension ofconversation whereas (arguably) pictorial stimuli might be more appropriate as a means for promoting speaking skills.

One of the considerations that prompted support for so many representational modalities was that prospective users of the system (foreign language faculty) hold different opinions about the best way to teach vocabulary. Some argue that native language words should never be used as stimuli and that pictures or motion video segments should instead be employed to stimulate recall of words in the target language. Other faculty are less adamant and themselves use a range ofrepresentations when they teach, including native-language equivalents. The system was designed to let lesson developers use the representations that they think are most effective.

Why IMPART Supports Both Multiple-Choice Response and Constructed Response Formats

Interference is a major obstacle to successful mastery of drill content. Salisbury, Richards, and Klein (1985) have suggested an operational definition of interference especially appropriate to computer-based drills: confusion between two stimulus-response situations. It might be concluded that situations likely to contribute to such confusion should be avoided. However, the essence of paired associate learning is that the student be able to distinguish an element from a field of candidates and match it with the presented stimulus. This is true whether the format is fill-in (in which case the element must be distinguished from all candidate elements stored in the student's memory) or multiple choice (in which case the element must only be distinguished from among the other options).

Should items be formatted to require response through multiple-choice options or through a constructed response ("fill in the blank")? We decided to make both formats available, in part because we had not researched the issue at the time of the decision.

IMPART automatically selects items from the working pool for presentation as multiple-choice options. In other words, it forces the student to distinguish the correct element from *n* listed elements where *n* (as set by the author) can range from 2 to 10. Thus, the author determines the degree of potential interference that will accompany each item. The reason for constructing items in this way is primarily practical: It eliminates the need for the lesson author to construct individual practice items and it permits students to respond instantly, without the problems associated with keyboard input.

We were also concerned that the use of constructed-response items would slow the drill sessions and reduce the number of items that could be practiced, especially for students with poor keyboard skills. Spelling errors further complicate the problem. It is fairly easy to build tolerance for minor spelling errors into CAI response evaluation routines. However, this is more difficult if the overall goal is to build an authoring system that can accommodate multiple languages governed by different rules for spelling and accents. In addition, it was also assumed that the processing required to do relaxed evaluations of responses might slow the system to the point where its performance would be unacceptable. This problem was judged too difficult to solve with available resources.

In the end (also as a matter of practicality), to accommodate the concerns of certain foreign language faculty who felt strongly about the matter, the system was designed to support both multiple-choice and fill-in responses. For fill-in questions, the system requires an exact response. In other words, IMPART considers a "nearly" correct answer, which differs from the correct response by as little as a single keystroke, to be incorrect.

A subsequent review of the literature suggested the issue is not settled. For example, Gay (1980) found that constructed response items resulted in equal or greater retention than did multiple-choice items, but Duchastel and Nungester (1982) found that constructed-response items do not necessarily lead to better retention that multiple-choice.

Why IMPART Uses Multiple-Choice "Foils"

Random presentation of multiple-choice options has a serious drawback. The reliability of multiple-choice tests is negatively influenced by a lack of plausible distractors. (In fact, much of the work of professional test item writers consists of inspecting item analysis data.) In IMPART, it is quite likely, therefore, that any given presentation of an item in multiple-choice format will contain options that can be easily discarded by the student as being very unlikely (on the basis of disagreement in tense or gender, for example). Furthermore, as items migrate to the mastery pool, the student works with a smaller and smaller set of items. To some extent this problem can be solved by specifying a larger number of options. However, we believe the use of foils is likely to improve the reliability of the system's quizzes and tutorials.

CONCLUSION

Like any tool, an authoring system adapts general principles to specific conditions and desired outcomes. If possible, the principles underlying an authoring system should be based on validated research findings. When designers are confronted by ambiguous or conflicting theoretical prescriptions relating to tool design, they should consider constructing (and testing) alternative prototypes based on contrasting capabilities. Another possibility (represented by IMPART) is to build a single prototype that operationalizes the conflicting prescriptions as alternative and contrasting capabilities of an integrated system.

Incorporating contrasting capabilities into prototypes offers two advantages. The first is that, in the absence of clear and unambiguous theoretical prescriptions, such prototypes can help to accommodate the personal preferences, hunches, and "styles" of lesson authors. The second advantage is that, properly conceived, a prototype with contrasting capabilities can support a series of related experimental treatments aimed at resolving the very ambiguities that underlie the design.

REFERENCES

Ambron, S. & Hooper, K (Eds.) (1988). *Interactive multimedia: visions of multimedia for developers, educators, and information providers.* Redmond, WA: Microsoft Press.

Anderson, J. R. (1980). *Cognitive psychology and its implications.* San Francisco, CA: Freeman.

Bonner, J. (1988). Implications of cognitive theory for instructional design: revisited. *ECTJ,* 36(1), 3-14.

Bower, G. H., & Hilgard, E. R. (1981). *Theories oflearning* (5th ed.). Englewood Cliffs, NJ: Prentice-Hall.

Brown, J. W. (1986). Some motivational issues in computer-based instruction. *Educational technology, 26(4),* 27-29.

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

Duchastel, P. C., & Nungester, R. J. (1982). Testing effects measured with alternate test forms. *Journal ofEducational Research,* 75(5), 309-313.

Edwards, E. A., & Siegal, M. A. (1985). *Teaching generalizations using a computer-based drill strategy.* Paper presented at the Annual Meeting of the American Educational Research Association, Chicago, IL.

Gay, L. R. (1980). The comparative effects of multiple-choice versus short-answer tests on retention. *Journalof'EducationalMeasurement,* 17(1), 45-50.

Gagne,R.M. (1970). *The conditions oflearning (2nd ed.).* New York, NY: Holt, Reinhart and Winston.

Hannafin, M. J. (1984). Guidelines for using locus of instructional control in the design of computer-assisted instruction. *Journal of Instructional Development,* 7(3), 6-10.

Jonassen, D. H. (1986). Hypertext principles for text and courseware design. *Educational Psychologist, 21(4),* 269-292.

Klein, J. D., & Salisbury, D. F. (1987). A *comparison of microcomputer progressive state drill andflashcards for learningpaired associates.* Paper presented at the Annual Convention of the Association for Educational Communications and Technology, Atlanta, GA.

Salisbury, D. F. (1988). When is a computer better than flashcards? *Educational Technology, 28(12),* 26-32.

Salisbury, D. F, Richards, B. F, & Klein, J. D. (1985). Designing practice: A review of prescriptions and recommendations from instructional design theories. *Journal ofInstructional Development, 8(4),* 9-19.

AUTHORS

Brockenbrough S. Alien is an Associate Professor of Educational Technology at San Diego State University, San Diego, California 92182-0311, where he teaches graduate courses in interactive video and instructional design.

Steven L. Eckols is an Instructor, Department of Educational Technology, San Diego State University where he teaches courses in educational computing and message design for interactive training systems.

ACKNOWLEDGEMENT: The authors wish to thank Alice Rogow, Yoshio Tanaka of San Diego State University, and Atsusi Hirumi, now at Florida State University, for their help in developing the IMPART system. IMPART was developed under the auspices of the Language Acquisition Resource Center and the Department of Educational Technology at San Diego State University under a California State Lottery Discretionary

Funds Grant. The opinions of the authors are solely their own and do not necessarily reflect the policies of the State of California or San Diego State University.