



**Australian Journal of
Educational Technology**

Educational software design: A literature review

Gennaro Pellone
Barton Institute of TAFE

The aim of this literature review is to outline the context of theory in designing educational software for vocational education and training, and to give an account of the significance of the "key ingredients" which constitute best practice.

With the advent of competency-based training as a system for vocational education and training, this decade has started to witness an increasing number of computers filtering into the classroom not only as tools for managing learning but more so as means of delivering self-paced instruction.

The role of many teachers and trainers is slowly but surely changing from the traditional lock-step giver of information into that of presenter, manager, facilitator of learning. Most of the old preparation and correction time is now being dedicated to developing educational and training resources, which includes educational software, suitable for a one-to-one delivery approach. By *educational software* it is meant the computer software program with which learners interact when being taught or being assessed by a computer, and by *design* it is meant the flow, the inclusion or omissions of certain key features responsible for the effectiveness of the program.

Pedagogy vs andragogy

Behind the design of any instructional technique there lie principles of learning. Over the years, educational psychologists have proposed many different theories primarily concerned with how the mind works in

relation to the learning process. These theories can be classified in several categories each emphasising certain aspects of the teaching-learning process more than others. Although "no current learning theory comes close to being complete" (Bork, 1985, p.86), for the purpose of this review, seen that the educational software in question is aimed at maturing human beings in vocational education and training, it is helpful to consider the assumptions about adult learners in contrast to how children learn.

Malcolm Knowles (1990, pp. 54-61) asks us to think about two sets of assumptions: one he labels "pedagogy" (concerned with the theories of youth learning) and the other "andragogy" (which relates to adult learning).

The pedagogical model is the one with which all of us have had the most experience. It is a teacher-directed model which gives the teacher the full responsibility for making all decisions about what will be learned, how it will be learned, and when it will be learned. The andragogical model, on the other hand, is a student-centred model which considers the self-directing nature of adults, their previous life experience, and their readiness to learn when they perceive a need to know or do something.

Both models are based on the following assumptions:

The need to know. Young learners only need to know that they must learn what the teacher teaches them if they want to get promoted. Adult learners need to know "why" they need to learn something before undertaking to learn it.

The learner's self-concept. Young learners have a dependent personality; the younger they are, the higher is their degree of dependency. Therefore, teachers are expected to take full responsibility for making decisions about what is to be learned, and how and when learning should occur. Adult learners, in contrast, have a self-directing nature and feel responsible for their lives and their decisions, including planning their own learning activities. They, therefore, resent and resist situations in which they feel others are imposing their wills on them.

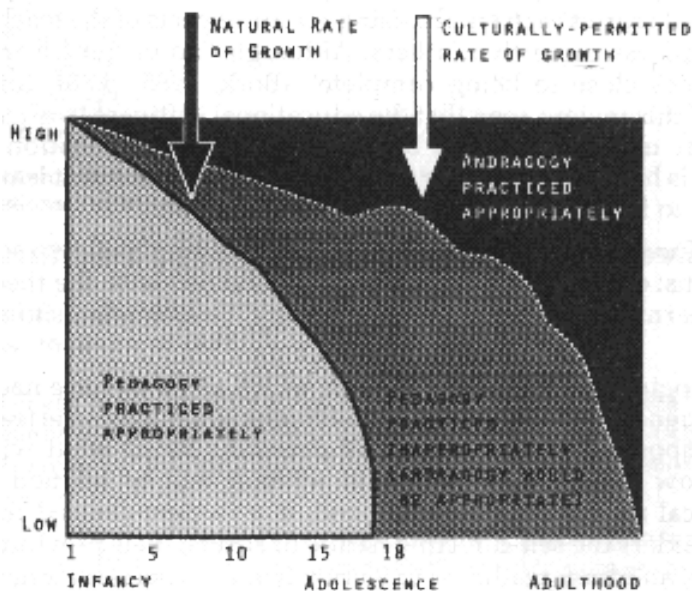


Figure 1: The natural maturation toward self-direction as compared with the culturally-permitted rate of growth of self-direction (Knowles, 1990, p.56)

The role of the learners' experience. A young learner's experience is too little to be of any value as a resource for learning. What counts in the classroom is the experience of the teacher and that of the textbook writer. Adults, on the other hand, enter the educational situation with a wealth of personal experience which can be a valuable resource to themselves as well as to others. Hence, a greater emphasis is placed on experiential techniques such as group discussion, simulation exercises and problem-solving activities.

Readiness to learn. Young learners become ready to learn when they are told by their teachers to do so if they want to pass in order to advance to the next grade level. Adult learners become ready to learn when they perceive a need to know or do something in order to perform more effectively in some aspect of their lives. Their readiness to learn is stimulated when they assess the gaps between where they are now and where they want and need to be.

Orientation to learning. Young learners have a subject-centred orientation to learning; they see learning as process of acquiring prescribed subject-matter content in a more or less logical sequence. In contrast, adult learners have a more life-centred or task-centred orientation to learning. They want to apply the new skills and knowledge as quickly as possible to real-life situations.

Motivation. Young learners are motivated to learn primarily by external pressures from their teachers and parents. Adult learners, while they are responsive to some external factors such as better jobs, promotions, and salary increases, the most compelling motivators are internal factors such as self-esteem, better quality of life, and greater self confidence.

Classroom instruction and educational software

A subscription to either the pedagogical or the andragogical model of learning carries with it certain implications for the design of educational software which is similar to preparing for any act of classroom instruction. Instruction being defined as "planned learning experiences" (Venezky and Osin, 1991, p.7): *planned* to mean deliberate but still leaving room for student control within the constraints of specified learning outcomes; and *learning* to mean an increase in accessible knowledge or ability. Similar to designing classroom instruction, therefore, the designer of educational software should make sure that:

- information is presented;
- the student is guided through initial use of the information;
- the student practices until familiarity is gained;
- student learning is assessed. (Alessi and Trollip, 1985, p.60-62).

The first element, presenting information, is to introduce students to something new. This is related to cognitive psychology which describes learning in terms of human information processing, as shown in the following diagram:

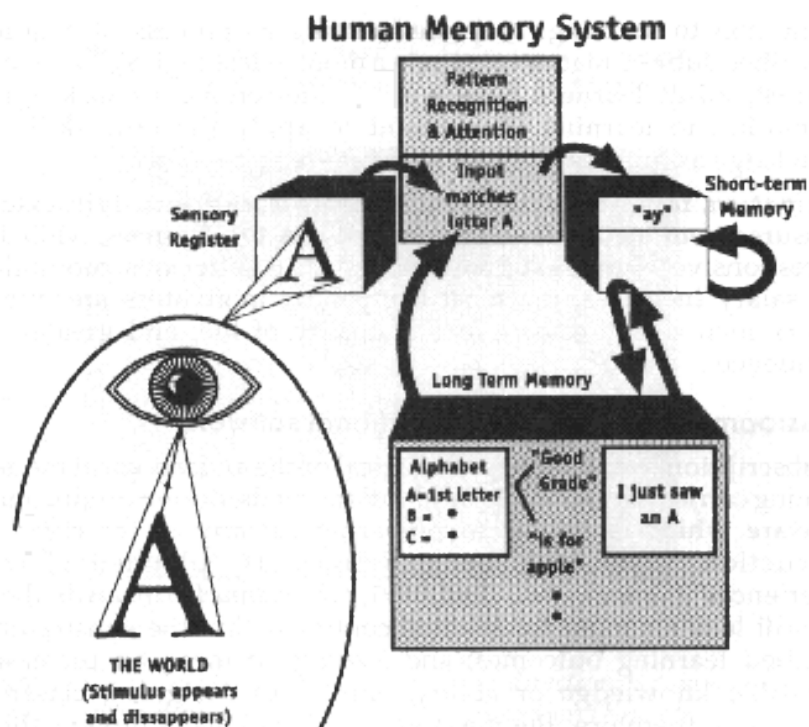


Figure 2: Model of human information processing (from *Human Memory*, 2nd ed, by Roberta L. Klatzky, in Criswell 1989, p.34).

Stimulation or information from the environment is input into the sensory register of the brain. Mental activities taking place in the brain of the recipient of the input include pattern recognition, short-term memory, and long-term memory. Input is encoded and stored in the brain in short-term or long-term memory. (Criswell, 1989, p.33).

These processes include motivation, selective perception, encoding, memory and retrieval. Perhaps, the most important element for the design of educational software is "memory" which is increased if the student's attention is captured, new information is connected thematically to information learned previously; and the information flow is kept high to avoid boredom (MacLachlan, cited in Criswell, 1989, pp. 36-37).

The second element of the process, guiding the student, is more interactive. Once the learner has been introduced to new information it is the teacher's responsibility to act as a guide through the concept or the process being taught. The guidance method may vary according to the nature of the material being presented. Nevertheless, the role of the teacher is to cut a clear path for the student by monitoring each student's performance and correcting any misinterpretation or distortion of the information. "Intelligent teaching is diagnostic teaching, that is, based on feedback about learner needs" (Venezky and Osin, 1991, p.71). In the classroom context the teacher asks questions which students must answer. According to whether those questions are answered correctly or not the teacher then takes some particular action. Similarly, when designing educational software the importance of asking questions frequently and making decisions based on their responses should not be understated for students to be guided through the learning process.

The third element, practicing until familiarity is gained, is to ensure that the student learns to perform the task efficiently with minimal or no error. The assumption here is that well-constructed practice makes perfect by establishing good habits. With repetitive practice the student will eventually perform quickly and effortlessly. In a classroom situation work sheets would be used. With educational software the repeated practice for sustaining, refining, or perfecting performance could take the form of "electronic work sheets" (Geisert and Futrell, 1990, p.86) containing intricate questioning and providing corrective feedback.

The fourth element, assessing student learning, is usually accomplished by means of tests. Assessment, whether in a classroom environment or through educational software, is important in order to gauge how effectively the instructional message was received and "to maintain consistent standards for the learning task" (Keller and Suzuki, 1988, p.429). It equips the teacher or the educational software designer with information regarding the extent to which learning has occurred in order to determine future instructional strategies.

In sum, the "educational" aspect of educational software design is enhanced if a similar concept to designing classroom instruction is applied: information is presented, students are guided through the learning path, repeated practice is utilised for perfecting performance, and finally assessing students to establish whether or not learning has taken

place. This does not mean that all these elements should be present at the same time in all educational software. More often than not educational software is used alongside teachers and other media to complete the instructional process, therefore, the elements are shared between them.

Types of educational software

Educational software designed to interact with learners when they are taught or assessed by a computer could be classified in four major types: tutorials, drills, simulations, and tests.

Tutorials. Tutorials are those programs "designed to replace the regular classroom teaching by providing full instruction of the learning material" (Shlechter, 1991, p.105). They are usually structured to present information in small hierarchical steps which incorporate theoretical explanations as well as practice of the new material. "All tutoring systems embody in one form or another a model of an expert (or competent performer) for the skills being taught" (Venezky and Osin, 1991, p.47). Generally, this expertise is made accessible to the learner in a series of "help screens" (Lillie et al., 1989, p.49) that learners can access at will. Help screens may provide assistance to solve difficult problems or suggest ways of attacking the problem by following some expert examples.

Tutorials usually begin with an introductory section and then follow this cycle:

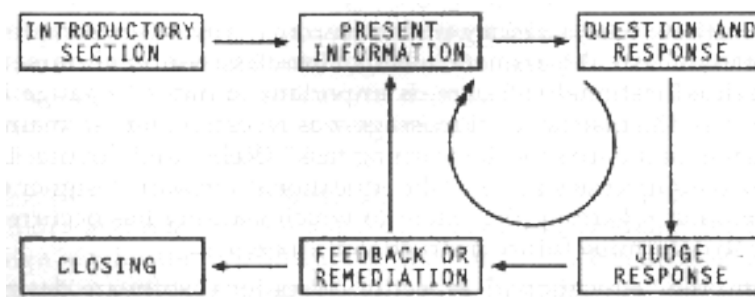


Figure 3: The general structure and flow of a tutorial
(Alessi and Trollip, 1985, p.66)

Information is presented and explained. Questions are asked to which learners respond. The program assesses the responses and, according to

their correctness, gives feedback and makes a decision as to whether or not to give some added information to improve comprehension and future performance. The cycle continues until the lesson is complete.

Drills. Drills, or drill and practice programs, help learners refine or enhance performance. They normally complement classroom instruction by reinforcing skills already learned. In such programs the usual job of the computer is repetitive and follows a distinct pattern. Geisert and Futrell (1990, p.85) explain that the routine is commonly quite simple:

- the learner is presented with a question or problem that corresponds to the target level of performance;
- the learner responds by typing in the answer;
- the computer evaluates the answer and provides feedback on its accuracy;
- if the answer is correct the learner is presented with another question or problem, if the answer is not accurate the learner is afforded the opportunity to try again.

Drills work well in a self-paced educational system giving the opportunity to fast learners to advance rapidly in the hierarchical levels of performance. "Assessment can be based on a succession of responses rather than on a single response. The mastery criterion might be a fixed number of successive correct answers, say three in a row, or it might be a certain percentage correct in the last set of ten exercises" (Steinberg, 1991, p.165)

Simulations. Simulations differ from both tutorials and drill and practice programs in that "the interactions of the learners are not responses to questions but rather decisions they make in a role-playing situation" (Lillie et al., 1989, p.39). They provide, with the aid of the computer and good colour graphics, instructional experiences in areas that are either too expensive, too dangerous, or too inaccessible to provide otherwise. In simulations the computer does not just present predetermined situations. "The strength of a simulation is the fact that a computer responds to student input. That is, the computer's responses depend on the choices students make" (Geisert and Futrell, 1990, p.96).

Alessi and Trollip (1985, pp. 162-171) divide simulations into four main categories: physical, procedural, situational, and process. A physical

simulation models some aspect of physical reality, such as an aeroplane cockpit, with which the learner must interact. Procedural simulations present a series of actions that constitute a particular procedure to be learned, such as diagnosing faults in automotive electronic circuits. Situational simulations represent human interactions with the environment or other people. Process simulations allow the learner to experiment with "what if" situations in a safe environment.

Tests. Tutorials, drills, and simulations present information and strategies to help learners achieve specific learning outcomes. Tests are used to assess this achievement. Usually tests are generated from a test item bank. "The concept of a bank refers to a very rich collection of questions, which provides an answer to the problem of avoiding repetition of the same questions by allowing for the construction of equivalent tests on the same subject" (Venezky and Osin, 1991, p.180). Scoring is accomplished by matching the learners response to that which has been recorded for that item in the test key. Therefore, the more open-ended the response becomes, the more complex the computer's task. True-false, multiple-choice, and matching questions are easily handled because the answer is finite and specific and can be expressed in a precise number or letter" (Lillie et al., 1989, p.104).

Designing instructional displays

The computer as an instructional medium offers learners self-pacing and, with its strong graphical capabilities, more dynamic representations than a blackboard or a whiteboard. However, a computer lacks the breadth of interaction that a teacher can provide. "Through tone of voice, facial expression, and other body language, a human teacher provides important extra information, both cognitive and affective, that a computer cannot express" (Larkin and Chabay, 1992, p.154). A computer is predominantly a visual medium and this factor advances a challenge when designing instructional displays. Following are some considerations which constitute best practice.

Screen layout. The layout of an instructional screen should be clear, uncluttered, and consistent. "The first thing to remember is that a computer display or screen is not a printed page" (Lillie et al., 1989, p.46). Printed pages present information all at once and have limited scanning control. Computer displays should be designed to selectively attend to important ideas by presenting information a little at a time.

The idea is not to reproduce a book on screen by filling each line margin to margin with text. Screens should be kept free of clutter by making a use of blank space; that is, by leaving borders, space between paragraphs, and so on. It is better to use "two or three sparse displays rather than one dense display to present material that need not appear all at once" (Larkin and Chabay, 1992, p.156).

Moreover, display screens should be consistent in format so that learners can get to know where to look for explanations, directions, and examples. Venezky and Osin (1991, pp. 214-216) suggest to organise the information into five "functional areas" which allow learners to develop adequate reflexes: main workspace, student answer, student command, system response, and status. Each functional area is identifiable by its location, its display characteristics, and its structure.

The *main workspace area*, in terms of space, would be the largest on the screen. It is where learners find explanations or questions. The main workspace may be structured differently according to the type of educational software. In tutorials, for example, a fixed subspace may be designated for diagrams or graphs. For drill and practice exercises the whole main workspace could be assigned to the question text while student options may be differentiated from the text area by colour, font, or graphic structure.

The *student answer area* is where student responses are echoed on screen. This area could differ for single key responses, such as answers for multiple choice questions, or short answer responses which are typed directly into given blank spaces.

A *student command* is usually represented by an icon or a function key. If function keys are defined for commands, no special functional area is required on the screen. If icons are used, they could be grouped in strips along the screen borders.

A *system response* area would be where either feedback is given to a learner's response, or a message is given in response to a command. In other words, it is an area assigned for acknowledgments, error messages, or requests for additional information.

A *status area* is where information, which is useful to the learner and the teacher, is given. Information could include student name or number, time elapsed since the beginning of the session, number of exercises presented and number of correct answers.

Text. Text serves two main functions in instructional displays. Firstly, it is used to present information about the topic, to ask questions, and to explain pictures and diagrams. Secondly, it is used to provide instructions for performing certain tasks, and to provide feedback on learner's responses. "There are no recipes for the creation of bright and clever explanations, but many good pieces of advice have been published that may help in avoiding the pitfalls frequently found in existing lessonware" (Venezky and Osin, 1991, p.217).

The following advice has been proposed by Alessi and Trollip (1985, pp. 75-78). New text should be added to a computer display by following the same convention of how people read; that is, left to right and from top to bottom. Sentences and paragraphs should be well formatted on the display. This means that lines should not end in the middle of a word and that paragraphs should not start on the last line of a display nor end on the first line of the next display; nor should they be squeezed into half of the display leaving the rest empty. Scrolling should always be avoided. Scrolling means adding text to the last line of a display forcing everything else to "jump" up a line making the top line disappear off the top of the display. Text should be, whenever possible, displayed in upper and lower case with appropriate punctuation. Slabs of text composed of all upper case letters is very difficult to read.

Menus. Menus provide learners with a direct and explicit selection procedure by choosing options from a list of topics. The major advantages are to "provide an obvious structure to the software" (Dumas, 1988, p.66) and "to 'break up' different sections of the program" (Favaro, 1986, p.128). When the number of options is very high, the usual strategy is to build a tree of menus. This will give learners a more holistic approach to the software by allowing them control over which learning path they choose for themselves. Menus should be designed so that they are easily operated; usually by pressing a number or letter corresponding to the chosen option. Two features need to be checked when designing menus. "First, the directions for making the choice must be clear, and the feedback must be informative should an inappropriate choice be made. Second, it should be easy to correct an incorrect choice. If, after typing the letter or number, the student has to press <RETURN> to activate the choice, a correction can usually be made by using the erase or arrow key before <RETURN> is pressed" (Alessi and Trollip, 1985, p.381).

Colour. Colour is an effective means of grabbing and holding student attention and many computers allow the use of many colour combinations. However, research literature "has very little to say on issues such as the "best" colour in which to display information and the best mix of colours" (Dumas, 1988, p.103). Nevertheless, the provision of colour to distinguish between different types of information and to highlight important or critical messages is preferred over monochrome screens. Venezky and Osin (1991, pp.222-223) explain that experienced designers work with compatible palettes, using usually not more than four colours per display. The combinations of colours should have enough contrast but should not "shout". A norm whose validity is clear from a physical standpoint, that is the strain placed on the eye muscles, is to avoid the combination of colours on the extremes of the visible spectrum, such as red and blue.

Graphics. Esther Steinberg (1991, pp. 142-143) explains that graphic displays can supplement understanding even when verbal text is straightforward, and can increase learner motivation and concentration. They also provide an alternative mode for learning and in many situations, graphic displays are the major means of transmitting information. For example, learning to read dials and gauges requires graphical representation in conjunction with verbal information. Because "graphics must support and enhance the written portion of the lesson" (Burke, 1982, p.64) a lot of consideration and attention should be given to their quality. That is:

- excessive detail should be avoided to simplify and demonstrate a point more clearly;
- consistency should be maintained to integrate each graphic into the total instructional message;
- more complex graphics should be broken down into their constituent parts and then built up in a series of overlay screens;
- each graphic should include labels and text so that learners can make informed comparisons and associations;
- all graphics should be in colour and in high resolution to display accurate and sharp images.

Animation. It has been documented that "well-designed lessons with animation have improved student scores more than presentations with graphics and text or text alone" (Back and Layne, cited in Richards and Fukuzawa, 1989, p.24). Animations capture attention more than text and provide dynamic explanations and demonstrations especially when

illustrating otherwise impossible situations such as following the current flow in an electronic circuit.

Including instructional enhancers

When designing educational software, the designer not only must consider learning principles and the effectiveness of visual displays, but also the inclusion of certain key elements to enhance the instructional process. These encompass:

Title screen. Instruction should begin by grabbing the learner's attention. A title screen is used "to create a receptive attitude" (Alessi and Trollip, 1985, p.67) and to indicate the nature of the lesson. Title screens could vary from a simple title to elaborate displays of multicoloured animated graphics.

Objective. "There is consensus among instructional design professionals that the presence of appropriate, measurable objectives improves the probability of a lesson's success" (Hannafin and Peck, 1988, p.17). A concise and accurate statement which includes what the learners will be able to do at the end of the learning program is highly desirable to focus the learners' attention more intensively on the attainment of the specified outcomes.

Entry level. "Research on human learning indicates that students will learn more if they can relate new information to what they already know" (R. C. Anderson, 1977; Adams & Bruce, 1980; Rumelhart & Ortony, 1977; in Alessi and Trollip, p.71). For this reason, a screen which specifies the entry level is an important component of the lesson especially when the new concepts to be learned require prior knowledge of a specialised nature.

On-line assistance. "Nothing can be more frustrating than being 'stuck' inside a program without any idea of what to do next" (Favaro, 1986, p.129). Therefore, a message should constantly be displayed on screen to provide learners with at least information about:

- how to get on-line help in case they get stuck and do not know how to proceed;
- how to access previous screens to review earlier information;
- how to exit the program at any stage.

Questions. Questions are an important element of the learning process and effective teachers use them regularly to keep students attentive to the lesson and to assess recall and comprehension. Steinberg (1991, pp. 103-105) highlights that questions may be presented before, during, or after learning has been completed, and that they serve different purposes at different points.

Questions presented *before learning* can serve as a placement device. If the answers reveal that a learner does not have the necessary entry level, an alternative might be suggested. If, on the other hand, a learner has already mastered some of the content of the lesson, then the program can allow to skip those sections. This saves time and avoids frustration and boredom.

When employed *during learning* questions serve several purposes: they can be used to gain and maintain learners' attention by piquing their curiosity; to help learners process information by focussing their attention on a critical feature of the material to be learned; and to foster remembering.

Questions presented *after learning* are useful in three ways: they can assess each learner's knowledge and skills; they can provide information about performance of an entire group; and they can assess the effectiveness of the lesson.

Branching. The educational path should take one of several alternatives depending whether learners master the material rapidly or vice versa. "Branching is the most prominent element that makes the computer a valuable tool for individualising instruction" (Schaefermeyer, 1990, p.9). Branching may take learners forward in the program, meaning that they skip some information if they demonstrate competence in questions and exercises; sideways to a remediation section if they identify unfamiliar areas of knowledge or record an incorrect response; or backwards to previous screens if they need to do so.

Remediation. "Remediation refers to the more extensive presentation of information for the student who is not learning the material" (Alessi and Trollip, 1985, p.119). In these circumstances, branching to restatements of the information in simpler terms, more examples, simpler graphics, and more practice exercises with simpler parts of the material should be provided.

Feedback. Feedback is the information given to a learner following correct or incorrect responses. It should always be aimed at being positive and corrective. Venezky and Osin (1991, pp.92-93) argue that most people like

to succeed and being told that they are correct may increase their self-esteem and their desire to learn more. Also, in case of an incorrect response, feedback should never demean the student. It should provide corrective information to improve performance.

Summary

To summarise, the heart of instructional computing rests on a foundation of effective teaching and instructional design principles. The educational software designer, therefore, must have an understanding not only of computer capabilities but also of instruction, classrooms and learners. Instruction requires the careful planning of learning experiences with each component of the plan interacting with the others, and well designed online instructional materials can help secure these experiences.

Learning is a dynamic process. This requires educational software to be designed in a way as to enhance the many instructional aspects of this process by including appropriate questioning and providing branching according to learners' responses. More so, quality instructional presentations using colour, graphics, and animation should be used to aid comprehension and retention. In the final analysis, the purpose of educational software design is to improve the quality of learning, and by employing principles of instructional design, stimulating visual displays, and relevant and corrective feedback the computer in the classroom can provide many effective learning experiences.

References

- Alessi, S. M. and Trollip, S. R. (1985). *Computer-Based Instruction: Methods and Development*. Englewood Cliffs, New Jersey: Prentice-Hall.
- Bork, A. (1985). *Personal Computers for Education*. New York: Harper & Row.
- Burke, R. L. (1982). *CAI Sourcebook*. Englewood Cliffs, New Jersey: Prentice-Hall.
- Criswell, E. L. (1989). *The Design of Computer-Based Instruction*. New York: Macmillan Publishing Company.
- Dumas, J. S. (1988). *Designing User Interfaces for Software*. Englewood Cliffs, New Jersey: Prentice-Hall.

- Favaro, P. J. (1986). *Educator's Guide to Microcomputers and Learning*. Englewood Cliffs, New Jersey: Prentice-Hall.
- Geisert, P. and Futrell, M. (1990). *Teachers, Computers, and Curriculum: Microcomputers in the Classroom*. Needham Heights, MA: Allyn and Bacon.
- Hannafin, M. J. and Peck, K. L. (1988). *The Design, Development, and Evaluation of Instructional Software*. New York, New York: Macmillan Publishing Company.
- Keller, J. M. and Suzuki, K. (1988). Use of the ARCS Motivation Model in Courseware Design. In D. H. Jonassen, (Ed.), *Instructional Designs for Microcomputer Courseware*. Hillsdale, New Jersey: Lawrence Erlbaum Associates, pp. 401-434.
- Knowles, M. (1990). *The Adult Learner: A Neglected Species*. 4th ed. Houston, Texas: Gulf Publishing Company.
- Larkin, J. H. and Chabay, R. W. (1992). *Computer-Assisted Instruction and Intelligent Tutoring Systems: Shared Goals and Complementary Approaches*. Hillsdale, New Jersey: Lawrence Erlbaum Associates, Inc.
- Lillie, D. L., Hannum, W. H. and Stuck, G. B. (1989). *Computers and Effective Instruction: Using Computers and Software in the Classroom*. White Plains, NY: Longman.
- Richards, T. C. and Fukuzawa, J. (1989). A Checklist for Evaluation of Courseware Authoring Systems. *Educational Technology*, 29(10), October 1989. New Jersey: Educational Technology Publications, Inc.
- Schaefermeyer, S. (1990). Standards for Instructional Computing Software Design and Development. *Educational Technology*, 30(6), June. Englewood Cliffs, New Jersey: Educational Technology Publications.
- Shlechter, T. M. (1991). *Problems and Promises of Computer-Based Training*. Norwood, New Jersey: Ablex Publishing Corporation.
- Steinberg, E. R. (1991). *Computer-Assisted Instruction: A Synthesis of Theory, Practice, and Technology*. Hillsdale, New Jersey: Lawrence Erlbaum Associates, Inc.
- Venezky, R. and Osin, L. (1991). *The Intelligent Design of Computer Assisted Instruction*. White Plains, NY: Longman Publishing Group.

Contributor: Gennaro Pellone has been working for over fifteen years as an Instructional Designer, Curriculum Services, at Barton Institute of TAFE. His main interest is in the design and implementation of vocational education and training. He is also a Lecturer in Training and Development at the Hawthorn Institute of Education, University of Melbourne, Australia. His address is Curriculum Services, Barton College of TAFE, Richmond Campus, PO Box 250, Richmond, Victoria 3121, Australia.

Please cite as: Pellone, G. (1995). Educational software design: A literature review. *Australian Journal of Educational Technology*, 11(1), 68-84.
<http://www.ascilite.org.au/ajet/ajet11/pellone.html>