

# Pedagogical Dimensions and the Evaluation of Multimedia Courseware for Computer Science

*Amruth N. Kumar & Claire Lichack*  
*Ramapo College of New Jersey*  
*505, Ramapo Valley Road*  
*Mahwah, NJ 07430-1680*  
{amruth,clichack}@ramapo.edu

## *Abstract*

*Fourteen dimensions have been listed for evaluating the pedagogy of multimedia courseware in literature[1]. Each of these dimensions is a continuum between two contrasting approaches. A courseware could qualify anywhere along the continuum for each dimension, depending on its features.*

*In this paper, we explore the following questions with respect to a select few dimensions:*

- *What are the features of a courseware module, that determine its position along a dimension?*
- *What are the implications of a dimension for the design and use of a courseware module?*

*We restrict our discussion to courseware for Computer Science topics.*

## 1 Introduction

An influential article on evaluating multimedia courseware was by Reeves and Harmon[1]. In their work, they have listed 14 pedagogical and 10 user-interface dimensions for evaluating interactive multimedia courseware for education and training.

Our focus is on the implications of the pedagogical dimensions on building and selecting multimedia courseware. In particular, we are interested in the features of a courseware module that determine its position along a given dimension. Listing such features helps us not only analyze a given courseware module but also build courseware modules which are pedagogically effective.

Following are the pedagogical dimensions for effective multimedia courseware, and the continuums corresponding to them, that we address in this paper:

1. Role of Instructor: Teacher-Proof Materials ↔ Teacher as Egalitarian Facilitator

2. Value of Errors: Errorless Learning ↔ Learning from Experience

3. Accommodation of Individual Differences: Non-Existent ↔ Multi-faceted

We will elaborate on these three dimensions in the next three sections. We have evaluated some of the other dimensions (Pedagogical Philosophy, Underlying Psychology, Instructional Sequencing and Experiential Validity) elsewhere[12]. In Section , we will illustrate our discussion by evaluating a courseware module we developed locally to teach iteration in C.

## 2 Role of the Instructor

This dimension deals with the continuum between the following two extremes:

- **Teacher-proof:** the courseware is designed to reduce or eliminate involvement of the teacher.
- **Egalitarian Facilitator:** the courseware is designed to include teacher/student interaction, with the teacher playing an integral role.

A teacher's involvement in a courseware module may be at various levels, starting with a teacher being incidentally present, to being a facilitator, or at the other extreme, being the intervening authority. Courseware in which the teacher acts as a facilitator or authority may be categorized as teacher-centered. We are considering the teacher's involvement only up to the role of an egalitarian facilitator in this discussion. The teacher may be a part of the instruction, feedback, navigation and/or use of the courseware (See Table 1). We will now examine the implications of teacher involvement on the design of courseware.

If the teacher is part of the **instruction** in a courseware module:

Table 1: Role of Instructor - Teacher-Proof Versus Egalitarian Facilitator

	Teacher-Proof	Facilitator
Teacher	Irrelevant	Discussant
Learner	Autonomous Enquirer	Participant
Material	Comprehensive	Complementary
Approach	Self-Instructing	Pedagogic Resource

- The accompanying tutorial or explanation need not be in-depth. The animation being presented need not be explained textually, the problem being solved need not be accompanied by textual explanations. In fact, such tutorials and explanations can be distracting and counter-productive if the instructor expects to periodically intervene with instruction during the use of the courseware. This would result in loss of stimulus-control.
- The courseware need not be self-contained in its content. It can serve more as an organizing sketch than a detailed description of the topics. It is understood that the instructor will intervene to flesh out the details as necessary. The courseware will complement regular teaching by providing animation, dynamic problem generation, and hyperlinks.
- The courseware may be limited to just the concept being presented, and leave the job of making the interconnections among the concepts to the teacher. The teacher provides the “big picture”, whereas the courseware concentrates on the local details.
- Problems may be generated more for illustration than for feedback and evaluation. The task of generating/selecting problems for testing may be left to the teacher.

If the teacher is part of the **feedback** loop in a courseware module:

- The courseware module need not solve the problems it generates. Even when it does solve a problem, it need not explain the solution in textual form. In general, the feedback need not be very detailed, since a live instructor is in charge of ensuring adequacy of feedback.
- The courseware can reduce record-keeping and evaluation of student performance on-line.

If the teacher is part of the **navigation** loop in the courseware:

- No particular sequence may be prescribed among the topics, i.e., the author need not sequence the topics in the courseware in terms of pre and corequisites for each module. This disarray facilitates discovery learning by its very lack of explicit organization. A teacher who intervenes in the role of a guide may introduce instructivism. In other words, a courseware module designed with minimal navigational guidelines may be used for discovery learning as well as for instructivist learning, which are two extremes of pedagogical philosophy in the design of courseware[1, 12].

If the teacher is part of **using** the courseware, such as when the teacher instructs on how and when to use the courseware vis-a-vis other experiences such as laboratories, assignments, tests, etc.:

- Design issues of the interface are not as critical - the teacher can compensate for any shortcomings in its interface. From the author’s point of view, this allows him/her to spend more time developing content than designing the GUI (Graphical User Interface)/presentation of the courseware. From the learner’s point of view, this allows the learner to focus on the content and be less susceptible to distractions from the GUI/presentation. On the other hand, if the courseware is to be used by the student in the absence of an instructor, its GUI must be well designed - to be both robust and clear.
- The instructor can enrich the courseware experience by relating the courseware module to other activities of a student such as laboratories and assignments. This permits the courseware modules to be less applied and less situated, since the instructor can be expected to supply the context of application during the use of the module. In the context, the instructor may also interpret observations/results for the students.

Teacher-Proof: This may mean that the courseware module is self-instructing (and hence, does not need the presence of an instructor), or, it is so designed as to be impervious to individualistic deliveries by instructors. The implications of this are that the module must be self-contained, well-designed and complete, both in its content and its presentation.

- If the courseware has to be complete in its content, the author must ideally anticipate all possible questions about the content, and attempt to answer them in an organized manner. This is admittedly hard, and requires the author to list the questions that tend to get asked time after time by students in a course.

Table 2: The Value of Errors - Errorless Versus From Experience

	Errorless	From Experience
Teacher	Evaluation	Evaluation + Feedback
Learner	Guided	Discovery
Material	Response-Restricted	Response-Unconstrained
Approach	Behavioral	Cognitive

The presentation of the material must be based more on scaffolding than on epistemological/ontological organization of the concepts, i.e., the module is based more on the mechanism of learning than on the structure of the knowledge. For example, consider a module on scheduling processes in Operating Systems. In the epistemological organization, the author may present the various algorithms first, and compare and contrast them later. In the scaffolded organization, the author may present the underlying differences first (time saved versus priorities enforced, etc.). In order to make the content easy to understand, the author may present one scheduling algorithm, let the user practice it, and then, present the others as successive improvements/changes to the first algorithm.

- The courseware would include elaborate instructions on using the interface/GUI. The help information would be more detailed, constantly available and context-sensitive.

### 3 The Value of Errors

This dimension deals with the continuum between the following extremes (See Table 2):

- Errorless Learning: the contingencies of instruction in the courseware are arranged to let the learners provide only correct responses.
- Learning from Experience: the courseware is designed to allow the user to commit mistakes, and learn from them.

Errors play a large role in learning by problem solving, as in Computer Science. Therefore, when we evaluate courseware, we consider both how much it allows a learner to commit mistakes and how it handles those errors. We will first consider the degree to which courseware allows the learner to commit mistakes. We will derive a taxonomy of courseware modules based on whether

a courseware module does or does not present the possible answers, and whether it presents only the correct answers or both correct and incorrect answers.

◊ We begin by considering courseware that presents the possible answers:

- The courseware presents both correct and wrong answers. Examples include true/false, multiple-choice and “Match the columns” questions. The ratio of the number of incorrect answers to the number of answers presented in these questions indicates the probability of errors that the questions allow:
  - True/false questions allow errors with a probability of 50%. “Match the columns” questions allow a greater probability of errors.
  - Multiple Choice questions allow a greater probability of errors based on the number of options. We could further classify them based on the choice of options they offer:
    - \* Exactly one option is the correct answer, the rest are wrong answers.
    - \* Many options are correct to various degrees with only one option being the optimal answer. E.g., the morning star is also called:

1. Venus	2. Mars
3. Evening Star	4. Jupiter
5. Answers 1 and 3	6. Answers 2 and 4

In this example, “Answers 1 and 3” is the optimal answer, although option 1 by itself is a correct answer, and so is option 3. Another example is a multiple choice question with “All the above” as an option.

- The courseware presents only the correct answers. This is the trivial case of presenting the facts. It cannot be used for illustration or evaluation, and will not be examined here.
- ◊ We now consider courseware that does not present the possible answers. Instead, it expects free-form answers. It may or may not provide clues to each answer, such as the range of values for the answer (“Your answer is in the range 1-100”) and the correct type of the answer (“Your answer is a character”).
- The courseware admits only correct answers. This is the case of errorless learning. The courseware poses a question and allows the user to enter only the correct answer, while refusing to entertain any incorrect attempts.
  - The courseware admits both correct and wrong answers. Here again, we could categorize problems

based on the available set of correct answers. The simplest case is one where only one correct answer exists. The other cases which are more challenging are:

- Multiple correct answers exist. The question may be worded to elicit all the correct answers, or only some of them. These questions provide more room for error.
- No correct answer exists. In this case, the question may or may not clearly indicate in its instructions that it may not have an answer:
  - \* A question which does indicate that it may not have an answer would be worded as: “If this is true, state why. If not, explain why not,” or “Indicate which states are traversed, if any.”
  - \* An example of a question which does not indicate that it may not have an answer is: “Evaluate the following expression in C:”

13.5    %    2.5

The modulus operator (%) is not applicable to real numbers in C, and hence, the answer is that the expression is syntactically invalid.

Questions such as this violate the expectations of the students regarding the existence of an answer, and could generate student frustration and elicit emotional responses which may interfere with learning.

The approaches in the above taxonomy vary widely in the design effort required to provide feedback:

- In errorless learning, since incorrect answers are not admitted, the underlying psychology employed for learning is behavioral rather than cognitive. (Underlying psychology is another dimension of evaluation of multimedia courseware[1] that has been discussed elsewhere[12].) Therefore, no explanation of erroneous responses is necessary.
- In the case where answers (correct and incorrect) are listed with the question, incorrect responses are limited to the options provided. Therefore, feedback explanations must be designed only for these incorrect responses which are limited in number.
- In the case where answers are not presented with the question, the range of incorrect responses could be so large that it may be impractical to account for all the possible scenarios when designing feedback. Pattern matching has been proposed as a solution to this problem[11].

The above three categories have been arranged in ascending order of the expense of design and development of response-analysis.

It is also important to consider what kind of feedback a courseware module provides to learners. Minimally, it reports whether the response is correct or wrong. In addition, it may consider the following:

- If the response is correct, it may explain why it is correct as a means of reinforcement. It may include a reference to the course text where the topic is discussed.
- If the response is incorrect:
  - It may display the correct answer.
  - It may also explain why the response is incorrect. It may include a reference to the course text where the topic is discussed.

Feedback may also be used to determine the future course of action (i.e., branching). Branching presupposes the pedagogical philosophy behind the courseware to be instructivist rather than constructivist[1]. The branching options are:

- Remediation, where a learner is returned to an earlier module (if the instructional sequencing is reductionist[1]), or returned to a less challenging level of questions in the current module;
- Repetition, where a learner stays at the current level of challenge in the current module;
- Advancement, where a learner “graduates” to the next module.

The branching may be based on one of the following models, listed in increasing order of complexity:

- Model of the **current session**, which includes quantitative summaries of the learner’s responses.
- Model of the **knowledge** being presented in the current courseware module. The model is used to hypothesize whether the learner’s response (which was incorrect) was for some other question in the module, and try to differentiate the knowledge being tested in that question from that in the current question. E.g., in a courseware module on Operating Systems, if a learner is asked a question on best-fit algorithm, but instead responds with a first-fit solution, the courseware module will attempt to explain the differences between the two algorithms.
- Model of the **learner’s knowledge**. This requires saving a history of all the sessions on all the courseware modules that the learner has completed so far.

Table 3: Accommodation of Individual Differences - Non-Existent Versus Multi-faceted

	Non-Existent	Multi-faceted
Teacher	Accommodation	Facilitating
Learner	Compensate	In concert
Material	Standardized	Customizable
Approach	No Options	Provides Options

In this case, once again, the model is used to hypothesize whether the learner’s incorrect response can be explained based on confusion about a concept or confusion among concepts. However, concepts may be considered not just from the current module but also all the earlier modules whose contents are reused in the current module. E.g., if a Data Structures module explains linked lists and arrays, and a Graphics module used later describes scan-line polygon-fill algorithm (which uses a linked list), an erroneous response from the learner in the Graphics module may be explained in terms of misunderstanding of linked lists if it cannot be explained in terms of misunderstanding of the scan-line algorithm.

## 4 Accommodation of Individual Differences

Multimedia courseware has an advantage over traditional teaching in that it can be customized to accommodate individual differences. Such differences may be personalistic (e.g., differences in learning styles, pre-requisite knowledge, aptitude and experience with the topic), affective (e.g., gender, age, race and culture-sensitive issues) or physiological (e.g., differences in eye-to-hand coordination, color acuity).

This dimension of courseware evaluation deals with the continuum between the following extremes (See Table 3):

- Non-existent: No provision is made for accommodating individual differences.
- Multi-faceted: Provisions are made to accommodate multiple individual differences, allowing sophisticated customization.

Here, we will consider some of the more common individual differences and how a courseware module can account for them.

- Physiological:

- Providing key-stroke alternatives to mouse selections.
- Providing the option to change the font (typeface and typesize) of text.
- Providing for color deficiency among users, which may be related to age, gender or other factors, by providing alternatives to color coding. Recommended rules of color usage[13] are followed such as that saturated blue should be avoided for text (to improve legibility), and spectrally extreme colors (such as red and blue) should be avoided to reduce eye fatigue.
- Learning styles - Different categorizations of learning styles may be considered and addressed:
  - Verbal versus visual learners[10]: Verbal learners, who prefer to hear the material, and see it in textual form may be accommodated by including rich text, hyperlinks, elaborate text layouts, scrolling textual commentary to accompany animations, and the audio medium. Visual learners may be accommodated by providing pictures, graphics, algorithm animation, visualization and videos.
  - Sequential versus global learners: Global learners may be accommodated by starting every module with an overview of the field, how the current module relates to the field, and an overview of the topics covered in the current module. Sequential learners may be accommodated by including at the end of every module a summary of what was covered in the module and how it relates to the other concepts in the field that may have already been covered.
  - Sensing (concrete experimentation) versus intuitive (abstract conceptualization) learners[10]: Sensing people may be accommodated by providing problem solving, animation and visualization. Intuitive people may be accommodated by providing a list of objectives, verbal descriptions, analysis, etc.
- Pre-requisite knowledge - Learners may arrive at a module with different levels of background knowledge. They can be accommodated by:
  - Providing hypertext links to panels of pre-requisite knowledge as a passive mechanism to “remediate” students.
  - Disallowing students from opening a module unless and until they have demonstrated proficiency in the prerequisite topics. This is a

more aggressive approach that results in a high degree of structure in the courseware.

- Providing hyperlinks to advanced materials on the web, including research results for advanced students. This may be through an automatic query sent to an intelligent web search engine.
- Aptitude:
  - Labeling problems by their degree of difficulty is one way to cater to students with varying aptitudes. However, this may affect the motivation of students, who may stay away from the harder problems even if they are quite capable of solving them.
  - The logical next step is to use adaptive testing: the better a student does in the early part of a test, the harder the questions (s)he is asked to answer in the latter part of the test.

## 5 Evaluation of a Sample Courseware Module

We applied the criteria discussed above to a courseware module developed at our college for the *Computer Science I* course. This module demonstrates the mechanisms of a counter-controlled loop in the C programming language. The module consists of a panel each for visualization, tutorial, status and user-input. (See Figure 1 for a schematic of the display.)

In the visualization panel, the courseware displays and identifies the components of a loop, viz., initialization, condition, updating and action statements. During animation, arrows are used to show the locus of control. The most recent arrow is indicated in bright blue, and the earlier ones are gradually faded out. The statement being currently executed is highlighted in a distinct color.

In the tutorial panel, parametrized text is displayed to explain every step in the animation. In the status panel, values of all the variables are displayed and updated in step with the animation. In the user-input panel, the user enters initial values for variables which enables him/her to explore different loop scenarios. The command buttons in the user-input panel provide the student with control of the animation.

We rated our module on a scale of 1 to 5 on each of the above-mentioned dimensions, as shown in Table 4. Our ratings are informal estimates; to the best of our knowledge, no precise yardsticks exist to place a courseware module along these dimensions. We rated our module a 2.5 in the dimension of the role of the instructor because it has both teacher-centered and teacher-proof elements

in it. It is teacher-proof because it includes a tutorial as well as visual cues (color, arrows) to assist the learner. On the other hand, the user interface is not robust and self-explanatory - it does not automatically direct user's input (by positioning the cursor), and does not clarify the sequence of actions expected of the user. Problems are generated for explanation rather than feedback and evaluation. The module is limited to the concept being presented, and relies on the intervention of the teacher to connect the concept to other concepts.

We rated our module a 2 in the dimension of accommodation of individual differences because the student has limited control over the appearance of the module. Although the student can control the speed at which the simulation executes, it is not possible to change the font style or size, the color of the arrows and color of the highlighted text. For some learners, the pre-determined colors may be difficult to see and may not capture their attention to the extent necessary. The module also restricts the student's control of cursor to mouse entries only. There is no provision for using hot keys instead of the mouse. All learners are assumed to have the same entry-level knowledge/skills and there is no provision in the module for remediation or enrichment via hyperlinks. Finally, no summary is presented at the end of the module.

We rated the module 1.5 in the dimension of value of errors because the simulation is constructed to accept only "reasonable" data before it executes, i.e., if the student does not enter any of the three values: initialization, condition, or update, the simulation will not execute. Further, it does not provide the student with feedback about the errors made in data entry.

## 6 Discussion

Our motivation in exploring the dimensions listed in literature[1] is to develop criteria to analyze existing multimedia courseware modules, and to build effective courseware modules of our own. We have attempted to map abstract dimensions of evaluation to concrete steps in the design of courseware that courseware developers can use to build effective modules. Future work in this project will include:

- analyzing the other pedagogical and user-interface dimensions listed in the literature[1].
- evaluating selected Computer Science courseware modules[6, 7, 8, 9] along these dimensions.
- continuing the evaluation of courseware we develop using the listed dimensions to ensure its effectiveness.

## References

- [1] Reeves, T. and Harmon, S.W., "Systematic Evaluation Procedures for Interactive Multimedia for Education and Training", *Multimedia Computing - Preparing for the 21st Century*, Idea Group Publishing, 1994.
- [2] *Authorware Pro*, Macromedia, San Francisco, CA.
- [3] Newell, A., and Simon, H., *Human Problem-solving*, Prentice-Hall, Englewood Cliffs, NJ, 1972.
- [4] Biermann, A.W., *Great Ideas in Computer Science: A Gentle Introduction*, The MIT Press, 1994.
- [5] Impagliazzo, J., and Nagin, P., *Computer Science: A Breadth-First Approach with C*, Wiley Publishers, 1995.
- [6] Boroni, C.M., and Eneboe, T.J., "Dancing with DYNALAB - Endearing the Science of Computing to Students", *The Proceedings of the Twenty-Seventh SIGCSE Technical Symposium on Computer Science Education*, February 15-18, 1996, pp 135-139.
- [7] Hartley, S.J., "Animating Operating System Algorithms with XTANGO", *The Proceedings of the Twenty-Fifth SIGCSE Technical Symposium on Computer Science Education*, March 10-11, 1994, pp 344-348.
- [8] Ingargiola, G., Hoskin, N., Aiken, R., Dubey, R., and Wilson, J., "A Repository that Supports Teaching and Cooperation in the Introductory AI Course", *The Proceedings of the Twenty-Fifth SIGCSE Technical Symposium on Computer Science Education*, March 10-11, 1994, pp 36-40.
- [9] Cowley, B., Scragg, G., and Baldwin, D., "Gateway Laboratories: Integrated, Interactive Learning Modules", *The Proceedings of the Twenty-Fourth SIGCSE Technical Symposium on Computer Science Education*, February 18-19, 1993, pp 180-183.
- [10] Felder, R.M., "Reaching the Second Tier - Learning and Teaching Styles in College Science Education", *Journal of College Science Technology*, pp 286-290, 23(5), March/April, 1993.
- [11] Bertrand I., "Free-Text Answer Analysis on the Web", *Revue Informatique et Statistique dans les Sciences Humaines (RISSH)*, Special Issue on "Internet and hypermedia systems", Joseph Denooz, Ed., 1996, pp 145-154, University of Liège.
- [12] Kumar, A.N., and Lichack C., "Pedagogical Evaluation of Multimedia Courseware for Science", *The Journal of Computing in Small Colleges*, Volume 12, No. 5, May 1997, pp 262-272.
- [13] B. Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 2nd Edition, Addison Wesley, Reading, MA, 1992.

Table 4: Evaluation of Our Courseware

Dimension	Score = 1	Our Score	Score = 5
Role of Instructor	Teacher-Proof	2.5	Egalitarian Facilitator
Value of Errors	Errorless	1.5	Experiential
Accommodation of Differences	Non-Existent	2.0	Multi-faceted

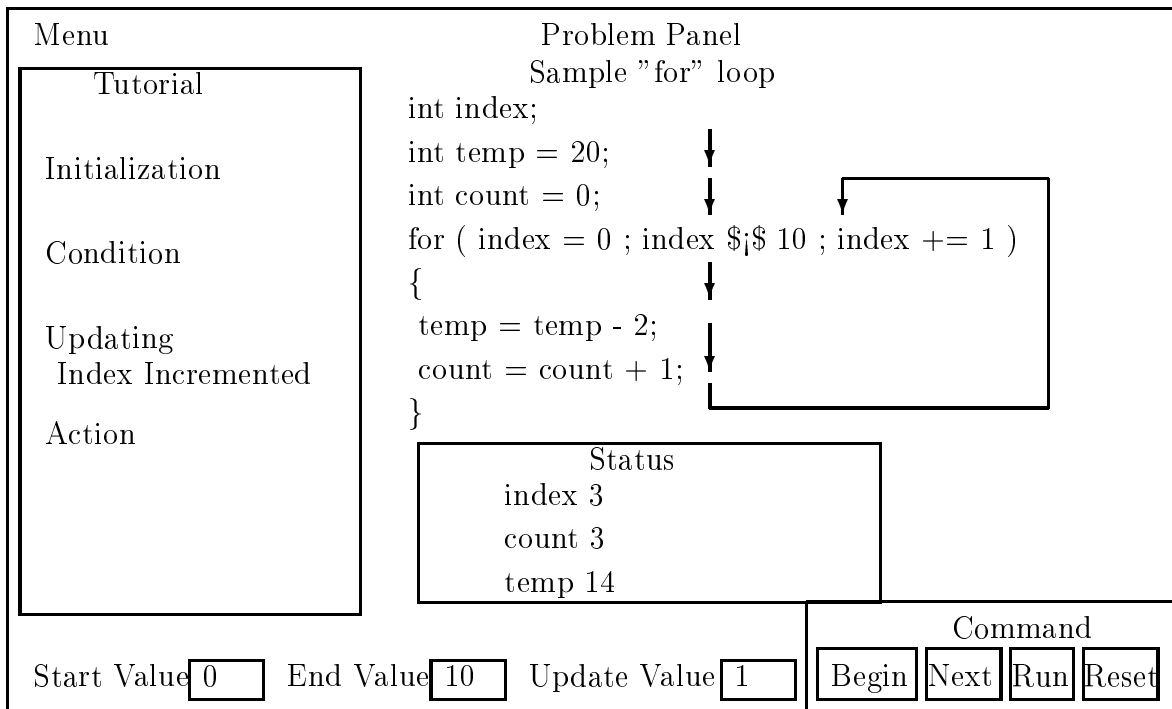


Figure 1: Schematic of the Courseware Display