DOCUMENT RESUME

ED 383 302 IR 017 157

AUTHOR	Hoover, Sheila J.; Abhaya, P. S.
TITLE	Instructional Design Theory and Scientific Content for Higher Education.
PUB DATE	95
NOTE	11p.; In: Proceedings of the 1995 Annual National Convention of the Association for Educational Communications and Technology (AECT), (17th, Anaheim, CA, 1995); see IR 017 139.
PUB TYPE	Reports - Evaluative/Feasibility (142) -- Speeches/Conference Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	Communication (Thought Transfer); *Computer Assisted Instruction; Computer Simulation; Computer Software Evaluation; *Content Validity; Higher Education; *Instructional Design; *Instructional Development; Qualitative Research; *Science Education; Surveys; *Teamwork
IDENTIFIERS	National Science Foundation

ABSTRACT

Computer-based instruction could have considerable impact on improving the quality of science education. Simulations and interactive problems provide a means for students to explore scientific concepts and experiment without the expense or hazard of using actual materials. This paper focuses on the instructional design process as it relates to the development of computer-based instruction for higher education. The Center for Interfacial Engineering Curriculum Development Project, funded by the National Science Foundation (NSF), is a partnership between engineers and educators to design and produce computer-based instructional modules. A survey was sent to other NSF grant recipients who were working on similar projects and also to an authoring discussion group on the Internet to obtain qualitative information on how they approach development, team member roles, whether or not team members have a science background, and, if not, how they communicate. Results indicate that: (1) it is important to clarify how much faith the content expert has in the instructional designer, and for each lesson to go through several iterations among team members; (2) the use of verbal explanations and sketching helps bridge the communications gap and discussions with nonscience team members help content experts to improve their teaching strategies; and (3) the software packages that were reviewed appear to have a lack of instructional design components. (AEF)

Title:

Instructional Design Theory and Scientific Content for Higher Education

Authors:

Sheila J. Hoover
Center for Interfacial Engineering
University of Minnesota

and

P. S. Abhaya
Center for Interfacial Engineering
University of Minnesota

Introduction

Instructional designers are expected to familiarize themselves with content from a variety of content areas. But what happens when the content is highly technical, scientific content and the instructional designer has little or no background in the area? Can instructional designers lend their expertise to curriculum development efforts at the postsecondary level in subject areas such as engineering, biology, physics, or chemistry?

Computer-based instruction for higher education science instruction appears to be growing. More science texts in university bookstores are being packaged with accompanying software. The National Science Foundation has funded several curriculum development projects around the country, and many of them are computer-based instructional multimedia projects. The Worldwide Web also provides access to several computer-based instructional modules for the sciences.

Computer-based instruction could have considerable impact on improving the quality of science education. Simulations and interactive problems provide a means for students to explore scientific concepts and experiment without the expense or hazard of using actual materials. One of the most powerful strategies for science education is visualization, the process of animating concepts on the computer screen that are too small or large, or too abstract to be viewed in real life. Visualization provides a means to bridge the gap between abstract processes and students' prior knowledge. According to Hiroshi Higuchi & Eric F. Spina at Syracuse University (1993),

> "Many undergraduate students have difficulty applying the basic principles learned in engineering problems, such as those encountered in design courses and experimental analysis. One reason for this shortcoming is that undergraduates often do not achieve a thorough understanding of the underlying physical principles in basic engineering courses. If, however, students are exposed to a visual representation of the relevant principle, and are encouraged to manipulate and study the image, then a lasting understanding of the principle may be reached - one based more on physical intuition and a "feel" for the phenomenon than on memorization."

Many of the computer-based instructional software packages for science on the market today do not use the computer to its full potential, or use the medium inappropriately, for example, simulations designed for the computer when the use of actual materials in a lab would be more appropriate. It is important for software development efforts in the sciences to focus on the computer as a tool to explain, elaborate, clarify, and visualize aspects of a lesson that cannot be explained as well on paper, or through a lecture.

Thus, computer-based instruction for the sciences is an area that could benefit from the expertise of instructional designers. However, there are several issues concerning the collaboration between content experts and instructional designers:

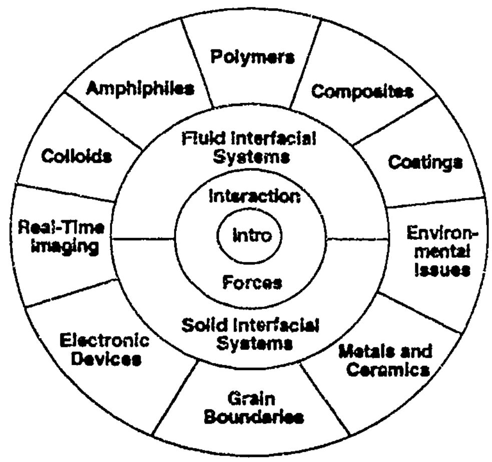
- Can instructional design theory be applied to higher education science content where the instructional designers don't understand the content?
- How familiar should an instructional designer become with the content?
- How can effective interactions be developed?
- How much time should the content expert be expected to put in?
- Who is a better source of content for creating interactive material -- the content expert or a novice in the field? Who is the most cost-effective source of information?
- Who could serve as an interface between the content expert and instructional designer?

This paper focuses on the instructional design process as it relates to the development of computer-based instruction for higher education. We examined (1) the instructional design process and lines of communication in our own computer-based project, (2) the instructional design process of personnel working on similar computer-based projects, and (3) existing computer-based software for higher education-level science content.

Center for Interfacial Engineering (CIE) Curriculum Development Project

Background

The Center for Interfacial Engineering (CIE) Curriculum Development Project began in June of 1993. Interfacial engineering is a new cross-disciplinary field that integrates research activities in aero, electrical, chemical, mechanical, and civil engineering, and chemistry. This curriculum development project is funded by the NSF and represents a partnership between engineers and educators to design and produce computer-based instructional modules. The modules have been used in conjunction with a textbook, Fundamentals of Interfacial Engineering.



Each module will have 5 tracks: (1) the main track provides the content, and is essentially a linear presentation of information. At certain points throughout the main track, the user may jump to other "tracks" to explore more in-depth information: (2) the theoretical track provides derivations of equations, (3) the experimental track provides simulations of experiments that explain how the information presented in the main track was obtained, (4) the application track provides real-world examples of the concepts in everyday life or industry, and (5) the interactive track provides problems and simulations that the user works through.

The development team currently consists of one primary and one secondary content expert, a full-time project manager/editor, a full-time instructional designer/graphic artist, and a half-time instructional designer/programmer. At the time of this writing, work on the main tracks is almost completed, and development of the theoretical and interactive tracks has just begun.

Development Process

When the project began, the content expert had completed the instructional analysis, and had developed storyboards that consisted of text and schematic figures drawn on transparency film. At the time, the development team consisted of one content expert, a full-time project manager/editor, one half-time instructional designer/graphic designer, and one half-time instructional designer/programmer. The initial development approach was to have a word processing specialist transcribe the storyboards into the authoring software, and then pass the module along to the project manager to edit the text and equations. Next, one of the instructional designers would program the module into the interface, and then give it to the other instructional designer to put in graphics and animations. As this system evolved, problems began to develop.

Content Experts and Instructional Designers

The working relationship between content experts and instructional designers does not have a reputation for being smooth, and there are many stereotypes about each other's competence. Stephen Lower, a professor of Chemistry at Simon Fraser University, in his paper, How to Make Computer-Assisted Instruction Fail (February, 1993), states

"The lessons themselves should have been designed by Education specialists, who are more likely to employ well-established response-reinforcement techniques. For example, Statement: "Fire engines are red." Question: "What color are fire engines?" B. F. Skinner used this technique with great success to train pigeons, who are evidently immune to boredom. This will turn away all but your most unimaginative drudges."

The instructional designers on the CIE project had heard that this attitude was common in the field. And initially this appeared to be true, because the content expert did not believe that instructional design theory was applicable at the postsecondary level, since most research on the subject dealt with instruction at the K-12 level.

As the instructional designers and content experts continued to discuss the development of the content, it became apparent that they were discussing the same instructional design principles, but using different terminology. For example, the instructional designers would talk about "test items" and "evaluation" and the content expert would refer to these concepts as "interactions." Consensus about instructional approaches improved once the team developed a common vocabulary.

Familiarity with the Content

Initially, the instructional designers believed that they were expected to become completely immersed in the content. The content expert knew that they would never become completely competent with the material, but still had the expectation that the designers would be able to take the content, and, from a brief explanation, be able to transform the storyboards into workable modules.

As the content expert and instructional designers worked together, they eventually realized that the instructional designers would never understand the content entirely, but could understand the structure of, or relationships within, the content. For example, the instructional designers could never understand the full nature of *inverted micelles*, but they could recognize that they represent one type of amphiphilic aggregate. As a result, the development process became a more collaborative process, and the linear development approach was replaced by an iterative one.

As the instructional designers spent more time working with the content, they became more knowledgeable, and better able to see relationships. Also, they became more familiar with the conventions that the content expert used in the storyboards. For example, when the content expert drew a box with an arrow, it represented a beaker, and an enlarged view of what was happening in the beaker on a molecular level.

Overlap of Roles

Developing a module requires each team member to have some familiarity with programming, graphics, editing, etc. Problems arose when team members moved into another's area of expertise. For example, the editor had to re-word some incorrect, elaborate text passages that had been written by other team members, the programmer had to rewrite unnecessarily complicated code, and the graphic artist had to recreate inaccurate graphics and animations.

This problem has not been completely solved, but it has been minimized. In most instances, the text is usually changed only with the editor's approval, interactions are worked out on paper first and the content expert and programmer work together to put them onto the computer, and graphics are either rendered as schematically as possible.

Time Demands

Module development came to a halt when there were unanswered questions about the content. The development staff had to wait (sometimes for weeks) until the content expert could find the time to explain. An organizational expert on staff at the Center was asked to examine the team's process and determine ways to make it more efficient. After talking with the team and drawing a graphic representation of the process, he calculated that it would take over 20 years to complete the project, because each iteration involved the content expert.

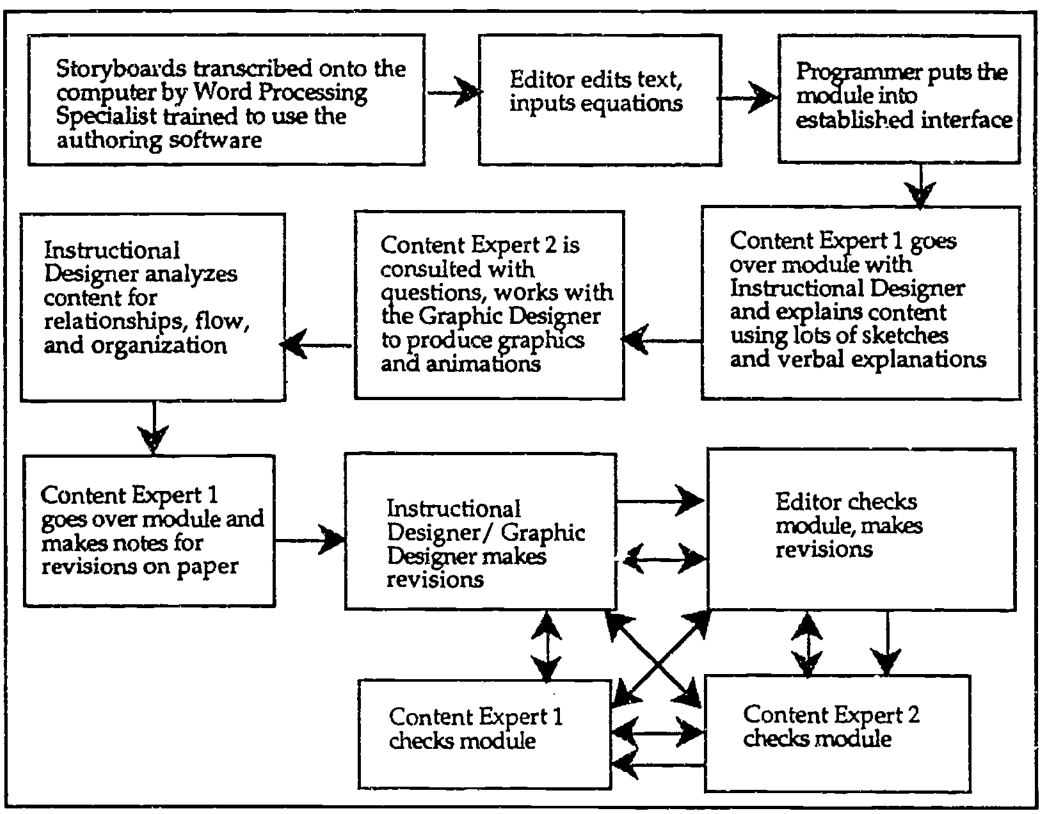
To solve this problem, a retired professor of chemistry was contracted part-time to serve as a secondary content expert. This solution had the greatest impact on the project. The professor was able to answer questions, explain processes, create charts and graphs, and edit content for accuracy. He provided a crucial link in the development process, as he had the necessary scientific background and the time to devote to the project.

Content Accuracy

The instructional designer/graphic designer would experiment with different ways of presenting the content. Most of the time the approaches were successful, and the rest of the team supported her creative license. However, on a few occasions, the content was altered to the point of inaccuracy, much to the distress of the project manager. The project manager asked the instructional designer to check with the content expert before changing anything, but the instructional designer felt that doing so would be excessively inhibiting.

After discussing the problem with the project manager, they agreed that the instructional designer could continue to experiment, but on highly technical content or on sections containing many equations, the instructional designer would work on a copy of that section, saving the original approach. When the project manager comes to that section, he can consult with the secondary content expert and make the final decision about which part to retain.

After addressing these problems, the development process evolved to the following:



Summary

For curriculum development projects in the sciences at the postsecondary level in which the instructional designers do not have a science background, it is important to clarify how much faith the content expert has in the instructional designer in terms of structuring content, and what their expectations are. It is also important for content experts and instructional designers to speak the same language with regards to the design of the instruction.

Each lesson must go through several iterations among team members, whose roles should be defined as clearly as possible. These iterations between the team members ensures accuracy of content, clarity of organization, appropriate use of the computer's capabilities, and ease of navigation. If the content expert does not have the time to complete several iterations, a secondary content expert, such as another professor or a graduate student, may be able to bridge the gap.

Other CBI Development Efforts

Survey

In order to learn about the design process of other computer-based development efforts in the sciences, we sent a survey to other NSF grant recipients who were working on similar projects at the postsecondary level. We also sent the survey to an authoring discussion group on the Internet. We were interested in qualitative information on how they approach development, the roles of team members, whether or not team members have a science background, and, if not, how they communicate.

Results

The majority of respondents were developing software to supplement a specific course, textbook, or lecture. They were initiated by college faculty who were the content experts for their respective projects. Most of their projects had been going on from 1 to 4 years. The scope of the projects was affected by the size and structure of the team. Projects that were developed by individuals were narrower in scope, dealing with specific lesson segments to supplement class lectures, and generally consisting of practice problems for the users. Team projects dealt with large units of instruction, sometimes at the curriculum level.

In order to examine the design process and the lines of communication in team projects, we divided the surveys into three groups: (1) content experts who work mostly on their own, (2) content experts who work with a team in which all members have a science background, and (3) content experts who work with a team in which not all members have a science background.

Development Approach

It seemed that the division of labor in a project where all members had a science background did not always take advantage of specialized skills. In some projects, the entire module was divided into lesson segments, and each member would work on a particular segment, doing all the writing, programming and graphics. For teams in which not all members had a science background, the development approach was more collaborative, although there was considerable overlap of roles. In the majority of projects where not all members had a science background, the content experts provided the initial storyboards and the subsequent development was done by the other team members, and the content expert continued to be consulted throughout the development.

Roles

When asked what roles were played by the team members, rarely was the term instructional designer used. The most commonly stated roles were content expert, graphic designer, and programmer. Most people filled more than one role, and it seemed that in particular programmers did a lot of the instructional design work, such as writing feedback, storyboarding content, etc.

Communication

As expected, when all team members had a science background, communication was not a problem. In teams where not all members have a science background, we were expecting respondents to cite communication as a significant constraint, but that was not the case. Apparently, team members have overcome communication barriers by using strategies that helped bridge the gap between what the content expert is trying to communicate and what the other team members understand. Most of them state that they use a lot of sketching and verbal explanations, and back this up with frequent checking and clarification of material by the content expert. In several surveys the respondents indicated that an added benefit to this approach was that, in the process of explaining things to the other team members, the content members learned to be more focused, and to make instruction more explicit.

Constraints

The most commonly listed constraints were financial and time. The majority of professors who responded stated that it was difficult to devote enough time to the development of computer-based instructional projects because of other demands, and because professors may not receive publication credit for software.

Summary

Many teams are working together successfully to design and develop science software, despite the fact that not all team members have a science background. Survey results indicate that the communication gap is not seen as a constraint. The use of verbal explanations and sketching helps to bridge the gap. In addition, content experts indicated that discussions with other nonscience team members helped them to improve their teaching strategies in the classroom. Results indicate that the role of the instructional designer in technical and nontechnical content areas is different, especially in the Analysis and Design phases (see Table 1 for a comparison).

Table 1. The Role of the Instructional Designer in Highly Technical vs Nontechnical Content Areas (assuming the ID has no prior knowledge with the content)

	Highly Technical Content Areas	Nontechnical Content Areas
Analysis	The content expert completes the needs assessment, identifies the instructional problem, and does the task analysis. The instructional designer's role is to ask relevant questions to guide the content expert *(for example, "What is the gap between what students should know and what they actually know? What are the goals, learner characteristics?" etc.).*	The instructional designer completes the needs assessment, problem identification, and does the task analysis.
Design	The content expert writes objectives, develops test items, and plans the instruction. The role of the instructional designer is to provide examples of instructional strategies that the content expert might adapt, and to ask questions that clarify the objectives, the content structure, the kind of learning that's supposed to take place, the kind of activities that will best facilitate learning, the graphics that go with the lesson, etc. *(for example, "How would you approach a lesson/concept? Problem first or description first? What is a good example of this relationship? Should the lesson branch out at this stage? If yes, where to?").*	The instructional designer writes objectives, develops test items, and plans the instruction. The content expert serves as consultant, and answers specific content-related and structural questions, clarifies terms, jargon, etc.

Table 1 (continued). The Role of the Instructional Designer in Highly Technical vs Nontechnical Content Areas (assuming the ID has no prior knowledge with the content)

Development	The content expert generates storyboards that establish the sequence of instructional events. The role of the instructional designer is to gain as much familiarity with the content as possible, to try to recognize relationships in the content, and to make sure that there is flow and consistency. The instructional designer and the content expert work together to make decisions about media.	The instructional designer writes the content, develops storyboards, develops a prototype, and goes back to the content expert for feedback.
Implementation	The instructional designer prepares necessary documentation and oversees installation.	The instructional designer prepares necessary documentation and oversees installation.
Evaluation	The instructional designer evaluates the materials based on the objectives and test items generated by the content expert. The content expert and instructional designer work together to make revisions.	The instructional designer evaluates the materials and makes necessary revisions. The content expert serves as a consultant.

Existing Science Software

We were interested in finding out whether or not existing science software for higher education demonstrated evidence of effective instructional design: (1) orienting information, (2) screen design, and (3) navigation, and (4) effective use of the media. We were not able to evaluate the actual content, since the software packages dealt with a wide variety of highly technical content areas. The software that we reviewed included a sample of computer-based modules that are currently packaged with science textbooks at the University of Minnesota bookstore, and science software packages that we obtained by word of mouth.

Results

All of the software that we looked at was intended to be used as a supplement to a textbook or lecture, in varying degrees. Some of the software consisted of complete instructional modules. Most of the software that was packaged with a textbook simply provided practice problems for particular segments of a lesson or course.

Orienting information

Many of the applications that we looked at consisted of a main menu that listed different sections that could be accessed by clicking on them with the mouse. Across all of the software that we reviewed, there were no goals or performance objectives, and very little feedback. The feedback generally consisted of "correct" or "incorrect" with no elaboration. The format generally consisted of an instructional unit followed by practice questions. The practice questions were generally multiple choice, and the majority of the questions required the user to input a correct answer before continuing. Some of the practice questions asked the user to input data, and these also required a correct number to be input before moving on. None of the modules contained an on-line glossary, and only one module had a help button. None of the modules provided page numbers, or any means to let the user know how much they had completed or how much was left.

Screen design

The software ranged from those that used effective, simple graphics and animations to those that were purely text-based. The software that was mainly text-based appeared to be better suited for paper-based presentation. For the software that incorporated graphics and animations, the font used was easy to read, but the on-screen text generally appeared to be excessive. In general, elements on the screen (text, graphics, navigational buttons) appeared in different locations on the screen in different sections of the lesson.

Navigation

The software ranged from that which presented the information in a linear fashion to software that allowed the user to jump to different sections, or to different segments entirely, such as problem-solving or experiments. However, across all of these different approaches, the majority did not have a means for the user to back up. In the software that provided a main menu, the users always had the option to jump back to the main menu, but not to back up within the section.

Effective use of the media

Aside from the purely text-based approaches, the majority of software packages that we reviewed took advantage of the computer's ability to visually demonstrate abstract concepts. The graphics and animations were simple, yet effectively demonstrated the concepts. Most of the interactivity of these modules was navigational, where the user selected the section of the lesson that they wanted to access. Some of the modules had simple interactions in which the user input data and clicked a button to see the graphic result. A few of the modules incorporated simulations in which the user could manipulate data in the context of solving a larger problem. Most of the interactivity was not incorporated into the instruction, but existed as a separate section.

Summary

There seemed to be a general lack of instructional design components in the software packages that we reviewed. Each package had at least one component of effective instructional design, but no single package demonstrated all. For example, one software package had good graphics and animation but didn't have any feedback, another package had very effective help buttons but didn't allow the user to move backward. Several of the applications were entirely text-based and would be better suited for paper presentation. All of the software lacked goals and objectives. Many of the modules had practice questions, but there was not much feedback and it generally consisted of "incorrect" and "correct," and most questions required a correct answer before continuing. Very few of the modules we looked at were interactive, and most of the interactions consisted of the user inputting data and pushing a button to see a graphic result, without showing the process.

Conclusion

Through visualization, simulations, and interactions, the computer has great potential for improving science education at the postsecondary level. Many currently available software packages on the market today do not seem to take advantage of the computer's capabilities, thus the application of instructional design principles to future computer-based development efforts in the sciences could help to produce more instructionally sound software.

It is possible for instructional design theory to be applied to higher education science content. The primary difference between technical and nontechnical instructional design approaches is that in technical content areas, the instructional designer's role is to guide the content expert through the analysis and design stages, and there are more iterations with the content expert throughout the entire development process.

It is not necessary for instructional designers to become completely immersed in the content, but it is important for them to become as knowledgeable as possible in order to ask relevant and guiding questions. It is important to develop common vocabulary about approaches to instructional principles. The communication process between content experts and instructional designers can be facilitated through the use of verbal explanations and sketches, and other professors or graduate students may help to bridge the gap between the knowledge of content experts and the backgrounds of instructional designers.

References

Hiroshi, H. & Spina, E. F. (1993). Improving undergraduate engineering education through scientific visualization. Syracuse University. ASEE Annual Conference Proceedings.

Lower, S. (February, 1993). How to make computer-assisted instruction fail. Simon Fraser University.