Pergamon

S0360-1315(96)00010-3

Computers Educ. Vol. 27, No. 1, pp. 15–22, 1996
Copyright © 1996 Elsevier Science Ltd
Printed in Great Britain. All rights reserved
0360-1315/96 $15.00 + 0.00

USABILITY AND LEARNING: EVALUATING THE POTENTIAL OF EDUCATIONAL SOFTWARE

DAVID SQUIRES[1] and JENNY PREECE[2]
[1]School of Education, King's College, Waterloo Road, London SE1 8WA, England
[*e-mail:* d.squires@hazel.cc.kcl.ac.uk; *fax:* +44 171 872 3182].
[2]Centre for People and Systems Interaction, School of Computing Information Systems and Mathematics,
South Bank University, 103 Borough Road, London SE1 0AA, England
[*e-mail:* preecej@vax.sbu.ac.uk; *fax:* +44 171 815 7499]

(Accepted 20 February 1996)

Abstract—The evaluation of the use of educational software must take account of usability as well as learning and, crucially, the integration of tissues concerned with usability and learning. Checklists, a very common approach to the predictive evaluation of educational software, do not take account of this integration. A review of two well known checklists illustrate this. A structured evaluation model known as the Jigsaw Model is described. Empirical evidence is provided to demonstrate that teachers' awareness of integration can be greatly improved by using the model. Copyright © 1996 Elsevier Science Ltd.

INTRODUCTION

At present usability is dealt in an *ad hoc* way during the evaluation of the use of educational software. Usability is typically conceived simply in terms of operating the software. There is no consideration of the implications of usability features for the use of the package to achieve educational goals. In this sense the integration of usability and educational issues is not considered. This leads to consideration of arbitrary usability features which may or may not be important to achieve educational goals.

As multimedia educational software with attractive and easier to use interfaces becomes available, evaluators' attention to usability may become even more limited. A feeling that interface problems have been solved may prevail. This would by unfortunate, as it would encourage the continued neglect of the relationship between usability and educational issues. Just because an interface is easy to use does not mean that it is designed appropriately from an educational perspective. There is an essential relationship between the two which must be addressed to ensure good educational software design.

There is little evidence that instructors consider the integration of usability with learning when they evaluate educational software. Consequently, it is not the case that simply increasing the general awareness about usability during evaluation is adequate. There is a need to help evaluators consider the way in which usability and learning interact.

Here the focus of attention is the predictive evaluation of educational software, i.e. the evaluation of software prior to its intended use. Predictive evaluation typically occurs when teachers are either planning lessons or making purchasing decisions. The most well known way to do predictive evaluation is by checklists. Here, two particularly well known checklists that have been developed for this are considered. The analysis illustrates the failure of the checklist approach to address the integration of usability and educational issues.

A model which attempts to integrate these two features helps teachers to focus on both learning and usability issues and the interaction of the two. It is known as the Jigsaw Model [1]. Two studies in which the model was tested with practising teachers are reported. The evaluation by these teachers were qualitatively different after using the model and showed a greater depth of awareness of integration. Some conclusions are drawn about the strengths and weaknesses of the model.

LEARNING AND USABILITY IN PREDICTIVE SOFTWARE EVALUATION

The purpose of educational software is to support learning in some way. Clearly the software design should take account of the way students learn and also provide good usability so that student's interactions with the software are as natural and intuitive as possible. For this to be so, the software must be consistent, predictable, reliable, responsive and provide appropriate help if things go wrong [2]. However, thinking of learning and usability as independent issues, which tends to be the usual way to consider them, leads to superficial evaluations of educational software. The design of an interface supports certain styles of interaction which may or may not support the intended learning tasks. In this sense there should be a synergy between the learning process and a student's interaction with the software — usability features should not allow the software to be efficiently manipulated, but should also be appropriate for the intended learning tasks. When this synergy occurs the use of the software can by thought of as 'integrated', in that a seamless union develops between the use of the software and the learning process.

When selecting software for use by students, teachers need to be aware of learning and usability and their integration. Unfortunately, while their training supports them fairly well in considering learning, most are not trained to think about usability. The integration of the two is seldom considered.

Teachers' ability to make the sound judgements necessary for selecting good software for their students is further hampered by practical concerns. They often do not have the opportunity to try and test software as thoroughly as they would like. At best they may be able to borrow or use software which has been purchased by a colleague or which is available for this purpose in a learning resource centre. At worst their decision to purchase or not may depend on reviews in magazines and descriptions on the cover of the packaging. While there is no way to support such a worst case situation, it is possible to support the best case situation with good predictive tools. Such tools need to identify the way learning is supported by the software, the ease or otherwise with which learners are able to operate the software, and the integration of usability features with learning intentions.

The usual way of doing predictive evaluation is through an evaluation checklist. The aim of such checklists is to provide a comprehensive set of questions dealing with both educational and usability issues so that the teacher is assisted in focusing on the key criteria to evaluate. It is a common approach dating back to the early days of educational software use, e.g. [3], but it is still popular, e.g. [4], with new lists appearing for current software environments such as CD-ROM based packages [5] and hypertext software [6].

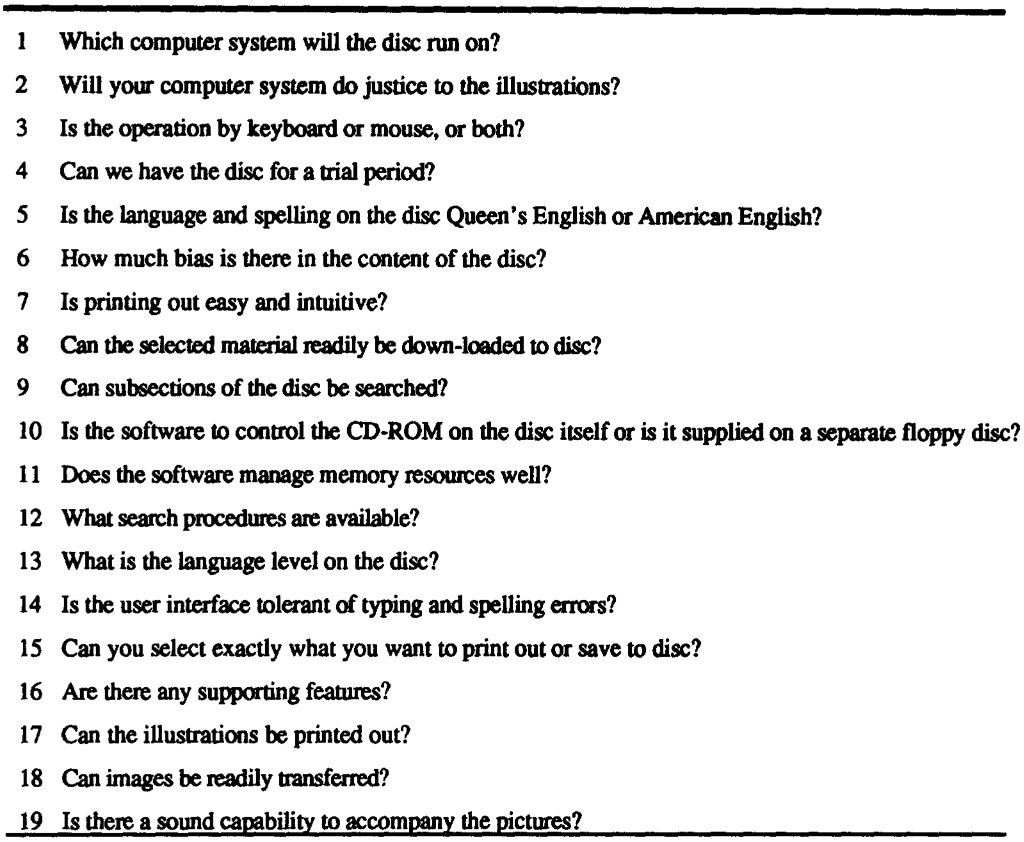
The ability of checklists to predict educational issues in all but a naive and superficial way has been questioned by a number of researchers (e.g. [7]). Learning issues are complex and highly context-sensitive: the questions that feature in checklists are too superficial to reflect this complexity. Usability issues do feature in these checklists but the approach to evaluating usability is typically an unprincipled, and somewhat arbitrary set of questions. At one level there is often confusion between specific usability issues associated with the educational application in question and the underlying operating system and even the hardware and peripherals. At a more fundamental level there is no consideration of interaction between learning and usability. Such problems appear in the following analyses of two well known checklists: a CD-ROM checklist developed by the British National Council for Educational Technology [5] and the American MicroSift Evaluators' Guide [8]. Both checklists have been developed for predictive evaluation.

NCET CD-ROM checklist

The questions proposed in checklist developed by the National Council for Educational Technology for evaluating CD-ROMs are shown in Fig. 1.

An inspection of the checklist shows that there are three explicit educational questions (5, 6 and 13), two pragmatic questions concerned with accessibility (1 and 4), and three questions are obviously technical in nature (2, 11 and 10). The remaining questions show that usability is interpreted in simple terms with no attempt to relate usability to learning:

- Is the operation by keyboard or mouse, or both?
- Is printing out easy and intuitive?
- Can the selected material readily be down-loaded to disc?
- Can subsections of the disc be searched?



1 Which computer system will the disc run on?

2 Will your computer system do justice to the illustrations?

3 Is the operation by keyboard or mouse, or both?

4 Can we have the disc for a trial period?

5 Is the language and spelling on the disc Queen's English or American English?

6 How much bias is there in the content of the disc?

7 Is printing out easy and intuitive?

8 Can the selected material readily be down-loaded to disc?

9 Can subsections of the disc be searched?

10 Is the software to control the CD-ROM on the disc itself or is it supplied on a separate floppy disc?

11 Does the software manage memory resources well?

12 What search procedures are available?

13 What is the language level on the disc?

14 Is the user interface tolerant of typing and spelling errors?

15 Can you select exactly what you want to print out or save to disc?

16 Are there any supporting features?

17 Can the illustrations be printed out?

18 Can images be readily transferred?

19 Is there a sound capability to accompany the pictures?

Fig. 1. NCET CD-ROM checklist.

- What search procedures are available?
- Is the user interface tolerant of typing and spelling errors?
- Can you select exactly what you want to print out or save to disc?
- Are there any supporting features?
- Can the illustrations be printed out?
- Can images be readily transferred?
- Is there a sound capability to accompany the pictures?

MicroSift Evaluators' Guide

The MicroSift checklist is shown in Fig. 2. Some of the information in the first section of this checklist is entirely objective, e.g. title, version, producer, and required hardware and software. Other items are more subjective, e.g. instructional purpose, instructional techniques, and estimated time of student interaction. The second section, entitled 'coursework evaluation', lists 21 items about content, instructional quality, and technical quality (Fig. 2). Evaluators are asked to score each item on a scale of strongly agree, agree, disagree, strongly disagree, or not applicable. They are also invited to mark the importance of the item as higher or lower. This section concludes with a request to state whether the evaluator would recommend the use of the package (possibly with modifications) and an invitation to describe major strengths and weaknesses and to describe the potential classroom use of the application.

A brief inspection of the items in these categories shows that both content and instructional quality are concerned with educational issues, while the terms in the technical quality section are concerned with usability. The decision to categorize items in terms of content, instructional quality and technical quality in itself implies a non-integrated approach. A closer inspection of the educational items shows a very limited consideration of educational and usability issues. Of the 14 items in these two sections only one item specifically addresses the integration between usability and

CONTENT

1 The content is accurate

2 The content has educational value

3 The content is free of race, ethnic, sex, and other stereotypes

INSTRUCTIONAL QUALITY

4 The purpose of the package is well defined

5 The package achieves its defined purpose

6 Presentation of content is clear and logical

7 The level of difficulty is appropriate to the target audience

8 Graphics/colour/sound are used for appropriate instructional reasons

9 Use of the package is motivational

10 The package effectively stimulates student creativity

11 Feedback on student responses is effectively employed

12 The learner controls the rate and sequence of presentation and review

13 Instruction is integrated with previous student experience

14 Learning is generalisable to an appropriate range of situations

TECHNICAL QUALITY

15 The user support materials are comprehensive

16 The user support materials are effective

17 Information displays are effective

18 Intended users can easily and independently operate the program

19 Teachers can easily employ the package

20 The program appropriately uses relevant computer capabilities

21 The program is reliable in normal use

Fig. 2. MicroSift checklist.

learning:

- Graphics/colour/sound are used for appropriate instructional reasons

The location of items on feedback and control over the rate of presentation in the instructional quality section implies an interpretation of these items entirely in educational terms.

Thus the checklist does not attend to the integration of learning and usability issues to any significant extent; only one of the 21 items is concerned with the integration between these two issues.

Concluding comments about checklists

This review considered just two checklists in detail — but the checklists are produced by well known educational bodies in Britain and the U.S.A. The analysis illustrates that the checklists do not deal with usability well nor do they integrate usability and learning. A predictive evaluation tool which teachers could use to evaluate educational software should encourage a consideration of the integration between learning and usability in achieving any learning task. Such a tool would help teachers to get good use from educational software. Obviously, what happens in the classroom is the most significant evidence of well designed and appropriately used educational software. Encouraging selection of good software is an important step on the way to achieving this. The Jigsaw Model has been developed with this aim in mind. In the next section we describe the components of the model.

THE JIGSAW MODEL

In common with current thinking, cognition is ‘distributed’ between users, the environment and learning artefacts, including computers, when learning takes place [9–11]. The distribution of cognition leads to learners constructing their own concepts which they use to learn. Adopting a distributed view of cognition leads to two consequences for predictive evaluation. Firstly, the learning supported by using the software should be interpreted from a constructivist viewpoint, and research into how students construct concepts in relevant subject domains should be referred to. Secondly, the use of educational software is context dependent. It is not possible to evaluate an educational software application predictively without reference to a perceived educational setting.

When software is used to support a learning task, in addition to developing concepts and skills in the domain of study, students also need to learn to operate the software effectively, in other words to do the operational task. The learning task comprises two main types of activities: (i) developing concepts and skills concerned with the specific topic being studied, and (ii) understanding the subsumed and prerequisite concepts related to the general domain of study. Successful learning will typically involve students in using and relating concepts from both types of learning activity. The operational task also comprises two types of activities: (i) interacting with the interface of the application, and (ii) interacting with the underlying operating system software, hardware and associated peripheral devices. There is also an important third area that the student needs to understand resulting from the integration of the learning and operational tasks, the integrated task. The ability to perform integrated tasks determines how effectively the software can support learning.

The learning, operational and integrated tasks that it are essential for the effectively using educational software are represented in the Jigsaw Model shown in Fig. 3.

At level 1 the learning tasks are simply thought of as individual learning concepts which are either specific to a given topic or which are related to the general area of study. Similarly, there is no link between the operational tasks concerned with the specific software application and the system operation tasks. The four tasks: specific learning tasks, general learning tasks, application operation tasks and general system operation tasks are all considered independently of each other. As we move from Level 1 to Level 2, integration within the learning and the operational tasks is considered, i.e. between specific and general learning tasks and between application and system operational tasks. At Level 3 integration between the learning and operational tasks is considered. This leads to the critical evaluation issue about how well the functions provided in the software and the usability of its interaction design match learners’ needs when using it. In other words, ‘how well has the software designer understood the learners’ needs and their model of how the software should function’ [12].

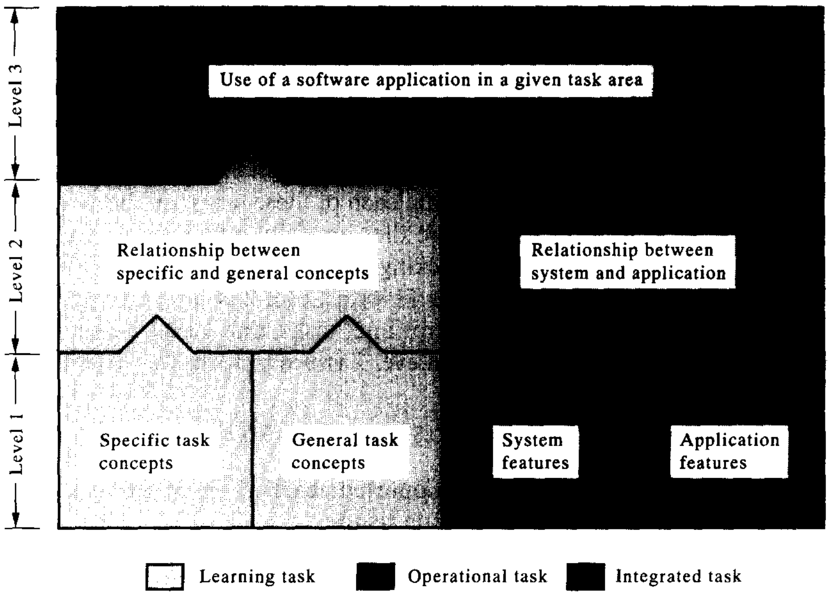

Level 3
Level 2
Level 1
Use of a software application in a given task area
Relationship between specific and general concepts
Relationship between system and application
Specific task concepts
General task concepts
System features
Application features
Learning task
Operational task
Integrated task

Fig. 3. Task levels within the Jigsaw Model.

Evaluator	Pre-JM	Post -JM
Instructor 1	Limited awareness: relationship between motivation and a gaming approach	Specific awareness: a concern that the search techniques provided by the program might encourage non-targeted enquiry rather than focused exploration.
Instructor 2	No explicit awareness	No explicit awareness
Instructor 3	Ability to generate " real time graphs [...] so that [the] relationship between qualitative and quantitative aspect[s] can be readily appreciated".	Use of the program to "time varying behaviour of factors involved in circular motion".
Instructor 4	Ability to draw angles in the 0 - 360 degree range in any position	Use of Logo to understand the concept of turning specific angles.
	Use of commands (procedures) to draw a variety of shapes.	Operational features limit usefulness - clearer graphics, prompts, and better accessibility are needed
Instructor pair 1	Complexity: levels of complexity could be provided to (i) cater for different abilities and ages, and (ii) to help in learning how to manage the different variables	Complexity: interface design related to coping with a number of variables at the same time: (i) skill levels for all the engineers should be displayed at the same time, and (ii) all job data should be displayed concurrently as opposed to in separate job specific windows.
Instructor pair 2	Decision making: difficult to display at the same time the necessary information to work out a decision.	Decision making: "Sometimes the program fails to present all necessary material for decision making on screen".
Instructor pair 3	General comment: the capacity to export data to a spreadsheet would provide opportunities for further investigation.	Information: (i) link between moving window positions on the screen and the display of sufficient information, and (ii) when changes to the times allocated to jobs were made it was not possible to adjust work schedules to accommodate these changes.

Fig. 4. Comments made by instructors before and after using the Jigsaw Model.

Figure 3 emphasizes that the learning and operational tasks can be considered as different amounts of integration. Integration of the components within each of these tasks and between them increases with progression up the levels until Level 3 where they are fully integrated to form the integrated task. The notion of a jigsaw puzzle with inter-locking pieces is a useful way of thinking about this integration. The figure shows how Level 1 issues fit with Level 2 issues through the relationships between the two learning tasks and the two operational tasks, and Level 2 issues fit with Level 3 issues through the relationship between the two Level 2 issues to form the ‘integrated tasks’.

STUDIES OF THE USE OF THE MODEL

To test whether or not the Jigsaw Model supported teachers in selecting software for their students two studies were carried out to investigate its use.

The first study involved four teachers studying an MA course module on design and evaluation of educational software. All regularly used software with their students and they had been taught to use checklists. Each teacher evaluator chose an application with which they were familiar and which was

Evaluator	Pre-JM	Post JM	Integration
Instructor 1	Yes	Yes	∂
Instructor 2	No	Yes	Δ
Instructor 3	No	No	
Instructor 4	Yes	Yes	∂
Instructor pair 1	Yes	Yes	∂
Instructor pair 2	Yes	Yes	
Instructor pair 3	Yes	Yes	∂

Fig. 5. Changes in the nature of comments made by evaluators.

relevant to their teaching. This software included an atlas of the solar system, a CD-ROM based database of video clips of motion sequences, a simulation of circular motion, and Logo. The first three of these applications could be described as adopting a 'black box' approach in that users are only exposed to a limited range of high level interaction options which are specifically geared to the learning task. In contrast Logo provides a low level approach in which users have more flexibility through exposure to general interaction options more tightly linked to operating system features.

The second study involved three pairs of teachers who were attending an advanced intensive course on the design and evaluation of educational software. All the evaluators evaluated the same package, (Priority) [13], which they had not seen prior to the evaluation and which may not have been relevant to their areas of teaching expertise. Priority consists of a simplified project management tool which enables students to schedule work assignments for four telephone engineers. Once a complete week's assignments have been specified the schedule can run to produce factual information about travelling times, wages etc. These values can be exported to a spreadsheet.

Method and data collection

In the first study, the teachers were asked to choose an educational software application with which they were familiar. They were then requested to spend approximately twenty minutes writing down comments about good and bad features of the application. A week later the teachers were taught to use the Jigsaw Model. After this session the teachers were asked to use the Jigsaw Model to evaluate the same software application they had considered in the previous week and to write down evaluation comments.

In the second study, the teachers were given a copy of the teachers' and students' notes which accompany the package and asked to evaluate the package using the MicroSift checklist. This evaluation period lasted for 90 min. Two days later the Jigsaw Model was described and explained to the instructors and its use was demonstrated. The teachers were asked to apply the Jigsaw Model to evaluate Priority and write down good and bad features of the program.

Data analysis

For both studies the evaluation issues that the participants identified prior to using the Jigsaw Model were classified according to whether they were concerned with usability, learning or the integration of the two. The classification was done by two researchers independently and controversial comments were discussed until consensus was agreed. Comments made about the integration of usability and learning both before (Pre-JM) and after (post-JM) the use of the Jigsaw Model are shown in Fig. 4.

Figure 5 shows the changes in the awareness of integration as an evaluation issue when the Jigsaw Model was used to assist evaluation. Inspection of Fig. 5 shows that awareness of integration was more developed in four cases and was introduced for the first time in one case. In one of the two cases in which awareness was not increased the evaluators already had a well developed awareness of integration before the use of the Jigsaw Model.

DISCUSSION

There is a clear need for tools which help teachers to select educational software. Checklists do not encourage a consideration of the integration between learning and usability in achieving any learning task. In this section the research findings of indicate how effective the Jigsaw Model is in supporting a consideration of integration.

The Jigsaw Model can help evaluators to develop an awareness of the significance of the integration of learning and usability issues. The majority of evaluators made comments after using the Jigsaw model that showed more integration. However, our findings also indicate that the use of the Jigsaw Model is most successful with software in which the interaction is at a low level, i.e. software in which system aspects are not deliberately hidden from users. Even so it is clear that the use of the Jigsaw Model can provide significant support for a principled approach to predictive evaluation of the use of educational software.

REFERENCES

1. Squires D. A comparison of learner and designer models in the use of direct manipulation educational software in the context of learning about interacting variables in photosynthesis. Ph.D. thesis, King's College London (1994).
2. Preece J., Rogers Y., Sharp H., Benyon D., Holland S. and Carey T., *Human-Computer Interaction.* Addison-Wesley, Wokingham (1994).
3. Heck W., Johnson J. and Kansky R., *Guidelines for Evaluating Computerized Instructional Materials.* National Council for Teachers of Mathematics, Reston, Va (1981).
4. Rowley J. E., Selection and evaluation of software. *Aslib Proc.* **45**, No. 3, 77–81 (1993).
5. Steadman S., Nash C. and Eurat M., *CD-ROM in Schools Scheme Evaluation Report.* National Council for Educational Technology, Coventry (1992).
6. Tolhurst D., A checklist for evaluating content-based hypertext computer software. *Educ. Technol.* **32**, No. 3, 17–21 (1992).
7. McDougall A. and Squires D., A critical examination of the checklist approach in software selection. *J. Educ. Comput. Res.* **12**, 263–274 (1995).
8. *MicroSift Evaluator's Guide for Microcomputer-Based Instructional Packages.* The International Council for Computers in Education, Eugene, Ore.
9. Brown J. S., Collins A. and Duguid P., Situated cognition and the culture of learning. *Educ. Research.* **18**, 32–42 (1982).
10. Honebein P. C., Duffy T. M. and Fishman B. J., Constructivism and the design of authentic learning environments: context and authentic activities for learning. In *Designing Environments for Constructive Learning* (Edited by Duffy T.M., Lowyck J. and Jonassen D. H.), pp. 87–108. Springer, Berlin (1993).
11. Salomon G. (Ed.), *Distributed Cognitions: Psychological and Educational Considerations.* Cambridge University Press, Cambridge (1993).
12. Norman D. A., Cognitive engineering. In *User Centred System Design* (Edited by Norman D. A. and Draper S. W.), pp. 31–65. Erlbaum, Hillsdale, N.J. (1986).
13. Flavell H., Gomberg M. and Squires D. *Priority.* British Telecom Education Service, London (1993).