28

The use of direct manipulation in educational software design

David Squires

School of Education
King's College London
United Kingdom

ABSTRACT

Direct manipulation is now the norm in software design. As a result this paradigm will feature significantly in the educational use of software. While the success of direct manipulation - providing the rationale for the development of easy to use, intuitive interfaces - is well rehearsed, it should be remembered that the aims of educational software go beyond simply presenting easy to use interfaces. The educational use of software has the grander aim of supporting cognition. In this paper a research study into human computer interaction issues, concerned with the design of a direct manipulation educational software package, is briefly described as a basis for considering the use of direct manipulation in the design of educational software. Four design paradoxes are identified which illustrate the potential conflict between educational issues and the use of direct manipulation.

Main conference themes: evaluation, software
Educational areas:
Study topics:
Secondary keywords: human computer interface, research

J. D. Tinsley et al. (eds.), *World Conference on Computers in Education VI*

INTRODUCTION

There are many positive reports of the use of direct manipulation interfaces. Users claim mastery of the system, competence in task performance, ease in learning the system originally and in assimilating advanced features, confidence in the capacity to retain mastery, enjoyment, eagerness to show the system to novices and desire to explore more powerful aspects of the system. Direct manipulation has been so successful that it is now a ubiquitous feature of software design.

Given the popularity of direct manipulation, students will inevitably use software based on this design rationale; either by default when generic software originally designed for use in commerce is used, or through the use of direct manipulation software specifically designed for education. However, the direct manipulation paradigm was not developed with educational issues in mind and, as such, its use in an educational context should be examined critically. In this paper a research study into the use of an educational direct manipulation application [1] is briefly described as a basis for discussing some implications of the use of direct manipulation as a design rationale for educational software.

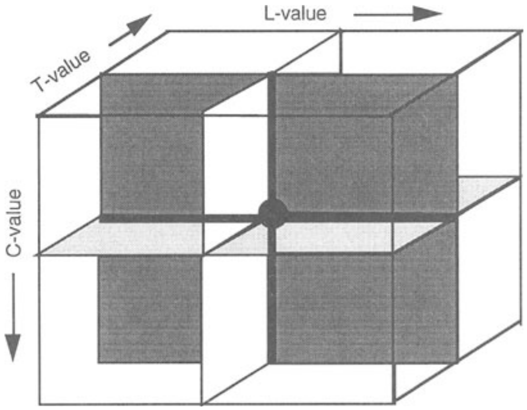

Fig. 1 The PSYNTH datacube showing an L-sheet, a C-sheet and a T-sheet.

A STUDY OF THE USE OF DIRECT MANIPULATION EDUCATIONAL SOFTWARE

The study was based on the use of *Bioview* [2], a *Windows* based software package which allows the exploration of the relationship between three interacting variables by the direct manipulation of a "datacube"; a three dimensional pictorial representation of data associated with the three interacting variables. The PSYNTH datacube was used in the study. This represents the rate of photosynthesis for three principal factors which interact to limit the rate of photosynthesis of a green plant - light intensity (L), carbon dioxide concentration (C), and temperature (T). Figure 1 illustrates the form of the PSYNTH datacube.

A datacube consists of three sets of commonly orthogonal datasheets with each datasheet corresponding to a fixed value of one of the datacube variables. In the PSYNTH datacube there are a set of L-sheets, each composed of rate values for a matrix of C- and T-values; a set of C-sheets, each composed of rate values for a matrix of L- and T-values; and a set of T-sheets, each composed of rate values for a matrix of L- and C-values. Thus data in the datacube are defined in terms of a sheet, row and column value. A set of sheets can be directly selected by clicking on the appropriate datacube face and a specific sheet can be selected by dragging a sheet through the datacube until the required position is reached. A row or a column can be directly selected by dragging a row or column across a sheet.

The results of directly manipulating a datacube can be represented in graphical and numerical form. For example, when a C-row is selected for a T-sheet, a graph of rate of photosynthesis plotted against light intensity can be depicted in a window. If the C-row is dragged across the T-sheet an animated graph will result showing how the graph changes for different C-values.

The research involved the use of *Bioview* by three pairs of Year 12 (age 16) students to answer set questions about limiting factors in photosynthesis. In each pair a student was asked to teach the other student to use *Bioview* to answer the set questions. These teaching sessions were video taped and the audio component of each video recording was transcribed. The Goals, Operators, Methods, and Selection (GOMS) model [3] was used to analyse the human computer interaction in these sessions. This is a well established model for human computer interaction [4, 5]. A goal is a symbolic structure which defines an aim and determines a set of possible methods to achieve it. Operators are elementary perceptual, motor or cognitive acts executed to change any aspect of the user's mental state or the task environment. A method gives a procedure for accomplishing a goal, and consists of a sequence of goals and operators, with conditional tests on the contents of the user's immediate memory and the state of the task environment. Selection is the decision by the user on which method to use.

The principal operators for the PSYNTH datacube are summarized in Table 1.

Table 1: Principal operators for the use of the PSYNTH datacube with *Bioview*

Operation	*L-value*	*C-value*	*T-value*
Select a sheet	s_sheet(L)	s_sheet(C)	s_sheet(T)
Move a sheet	m_sheet(L)	m_sheet(C)	m_sheet(T)
Select a row	s_row(L)	s_row(C)	s_row(T)
Move a row	m_row(L)	m_row(C)	m_row(T)
Select a column	s_col(L)	s_col(C)	s_col(T)
Move a column	m_col(L)	m_col(C)	m_col(T)
Inspect a rate/ variable graph	inspect_graph(r/L)	inspect_graph(r/C)	inspect_graph(r/T)

The goals were provided by the set questions. For example, the first question asked the students to find the optimum air temperature for the maximum rate of photosynthesis when light and carbon dioxide were not in limited supplies. A possible method for answering this question consists of four 'unit-tasks':

- select a T-sheet;
- maximize the C-value by locating the C-row at the maximum C-value position by executing an m_row(C) operation;
- inspect the resulting rate/L graph to determine the L-value corresponding to the maximum rate of photosynthesis; and
- vary the T-value by executing m_sheet(T) operations, and observe the changes in the rate-value corresponding to the maximum L-value shown on the resulting instances of rate/L graphs to determine the optimum temperature.

This method can be represented as the following expert 'action string' (with unit-tasks delineated by '||' markers):

|| s_sheet(T) || s_row(C) | m_row(C) || inspect_graph(r/L)
|| repeat [m_sheet(T) | inspect_graph(r/L)]

Action strings were written for the human computer interaction shown in the video records. A comparison between the observed action strings and corresponding expert strings allowed a comparison to be made of learner and designer mental models, leading to three findings. Firstly, the state of the system was very influential in the selection of methods. For the three sessions there was a total of 29 instances of method selection. In 18 of the 29 observed instances of method selection the selection was influenced by the currently selected sheet.

Secondly, the learners needed informative confirmation of the effects of their actions for a smooth interaction with the software. When the outcomes of direct manipulation were unexpected or unintelligible, goal directed cognitive behaviour was interrupted, compromising efficient task completion. This was illustrated by the effect of the appearance of null graphs corresponding to a row or column of zero values. Typically learners responded to such graphs by executing a series of panic operations, such as repeatedly connecting an already active graph window.

Thirdly, learners misunderstood the relationship between the state of the system and the representation of this state by the display. In particular there was a widespread misconception of the relationship between direct manipulation of the datacube and related changes in variable values. The values associated with the datacube are lexically scoped, that is, when the datacube is directly manipulated only the values associated with the currently selected datasheet are changed. Most of the learners assumed that the values were dynamically scoped, that is direct manipulation of the datacube would result in changes in the value of the variables for all three datasheets.

These findings suggest four design paradoxes for direct manipulation educational software: a black box paradox, a control paradox, a display paradox and an interaction paradox.

The black box paradox

The black box paradigm for software design assumes that the user does not need to understand how the system works. This approach is epitomized by direct manipulation, indeed the influential Model World Metaphor for direct manipulation [6] encourages users to regard their interaction with the machine entirely in terms of the interface metaphor. In this sense the metaphor is the system.

However, there is a penalty for the inherent simplicity of this approach. The learner is now completely reliant on the accuracy of the representation provided by the designer. Even if the representation is accurate, learners have to assume that the designer's model of the system is the same, or at least consistent at some level, with their model. This may not be true, as illustrated by the difference between the designer's structural model of Bioview based on lexical scoping, and the learner's model based on dynamic scoping.

Thus, it may be appropriate to adopt a contrary approach which makes a system model representation easily available to the learner. Du Boulay, O'Shea and Monk illustrate this approach in their concept of a glass box model of the computer [7]. The glass box model presents a relatively simple model of the structure and function of the machine which is adequate to understand the task in question. Providing a datacube values window in Bioview would follow this approach.

If users are unaware of the system model it is essential that the interface design is entirely consistent with the task model they hold, as any unexpected results can be only reliably interpreted in terms of the task knowledge they possess. Scoping in Bioview highlights the issue of the need for task based consistency in direct manipulation interfaces. In this case lexical scoping is consistent with the system, and dynamic scoping is consistent with the task. Conventionally, consistency tends to be interpreted in system terms [8]. However, as evidenced by the mismatch between learner and designer models of scoping in Bioview, the significance of an interface which is consistent in system terms can be lost on the user, and may in fact induce problems.

Thus it seems that the black box paradox offers two design alternatives — if you make human-computer interaction simpler by representing the application as a black box do not make its use harder by making the interface consistent with the system; if you make human computer interaction harder by requiring the user to have a structural model of the system, make its use simpler by providing a clear representation of the system at an appropriate level.

The control paradox

Giving the learner control to choose aspects of application functionality, and the way in which they are used, is commonly regarded as a desirable feature of educational software. Learner control implies that the learner's model should take priority over the designer's model, which suggests that the designer's model is based on the perception of a malleable software environment which learners can alter to suit their needs. Necessarily, the designer must make value judgements as to what level of control to provide; judgements which will be determined by the designer's model of the learner's model based on anticipated interaction problems and intended educational outcomes.

The issue of the delegation of learner control is critical to the direct manipulation paradigm. The success of a direct manipulation interface depends on the authenticity of the directness of the interface. Directness will only be authentic if the user feels totally in control of the software. If the designer patronizes the learner by assuming some of this control the directness of the interface will be reduced. Thus, while there may be compelling reasons for mitigating the effects of control by using techniques such as hiding and blocking functionality, the associated delegation of control to the designer can seriously compromise the validity of a direct manipulation interface. This dilemma is illustrated by the need for "action confirmation" observed during the use of Bioview. Learners had the freedom to produce null graphs, and in this sense they had total control over the system. However, as adduced by the number of consequent errors, this caused problems. The effect of null graphs could be mitigated through hiding, blocking, or dimming rows and columns corresponding to sets of zero values. The price paid for the positive benefits of mitigation would be a compromise in the authenticity of the model world metaphor presented by the datacube, that is a compromise in directness.

Hence the control paradox: Direct manipulation typically gives learners extensive control but potentially exposes them to the perils of ambiguous and misleading feedback; mitigating the effects of this misleading feedback results in a compromise in the validity of direct manipulation.

The display paradox

The widespread adoption of direct manipulation has required a conception of the role of the display in terms of the inter-referential nature of input and output [9]. Thus appropriate display design is an essential factor in the development of appropriate functional models of the system. Hence both the designer's and the learner's model will incorporate some consideration of the effect of the screen display, and the designer's model of the learner's model will need to account for any mismatch between the designer's and learner's perception of the display.

Making interaction easy in terms of a direct manipulation display implies efficient and economical manipulation of screen images. In turn, this implies that users should be required to make minimal changes to the display to achieve commonly desired effects. This implication is confirmed by the extensive method truncation, due to the use of methods based on the currently displayed datasheet, observed during the use of Bioview. Thus the state of the display was instrumental in the selection of methods. Such influence is not always beneficial; it may be that the display implies a method which is either sub-optimal or inappropriate, and the criterion for economical use has improperly constrained interaction.

If a direct manipulation interface is going to help easy use of a software application, it should guide the user to form functional models which will enable operations to be efficiently executed. However, the most economical choice of operations may not correspond to the functional models held by experienced users. This is the first form of the display paradox. The second form relates to initial use of an application. Again the display should guide the learner to choose appropriate actions, but at this level of expertise guidance can only be given in simple terms which link operators directly to goals. Hence the second form of the display paradox — indicating the superficially appropriate choice of an operator may lead to the unconscious adoption of an inappropriate method. The designer's functional models will take cognisance of the potentially misleading influence of direct manipulation screen displays, but the learner's functional models will typically not do this. Thus the designer's model of the learner's model must recognize these misleading effects, taking account of the level of expertise of the learner.

The interaction paradox

As Norman pointed out, the introduction of a cognitive artefact will change the user's perception of the task through a distribution of cognition between the user and the artefact [10]. Thus, while the actual task itself may remain the same, the responsibilities of the user will change. The way in which cognition is distributed between user and artefact will depend upon the nature of the interaction between user and artefact. Hence the design of the interface between artefact and user is critical in determining how the inherent functionality of the artefact affects the user's personal view of the task.

The comprehensive model world metaphor invoked by direct manipulation interfaces implies that the user's view of the functionality of a computer-based environment will be formed exclusively in terms of the direct manipulation techniques afforded by the interface. Therefore, the direct manipulation operators provided by the interface need to support effective management and manipulation of the interface, and the specific requirements of the task in hand. Hence, the use of direct manipulation has metaphorical implications for both the computer-intrinsic and the IT-applicational domains identified by Birnbaum [11].

The observed use of the direct manipulation of the datacube provided by Bioview illustrates how the nature of the direct manipulation techniques offered by an interface can affect the learner's view of a task. The value of only one variable can be varied at a time by direct manipulation, which may explain the use of an isolation of variables strategy [12] by the learners, rather than the more appropriate multi-variable approach. Thus, while the datacube direct manipulation operators (m_sheet and m_row/col operators) provided an intuitive way of manipulating the datacube, they did not relate adequately to the task related requirements. In this sense, direct manipulation of the datacube successfully attended to the computer-intrinsic requirements of the learners, but failed to adequately address IT-applicational needs of the learners.

An interface which is successful in computer-intrinsic terms and relatively unsuccessful in IT-applicational terms indicates that there may be a conflict between designer and learner functional models. As noted earlier the designer's model is often based on a system based perception of consistency, while the learner's model tends to be formed in terms of the task requirements. This implies that the interface has been designed primarily with system related consistency in mind. Thus the persistent dilemma of system as opposed to task based consistency arises again, this time in the form of the interaction paradox. As the demand on the learner of coping with computer-intrinsic features is mitigated through direct manipulation consistent with the system, the direct manipulation techniques afforded by the interface tend to become progressively less task consistent, leading to the possibility of inappropriate application of the techniques.

SUMMARY

Direct manipulation is now the norm in software design, and as such it will feature strongly in the educational use of software. However, it should be remembered that this paradigm was developed with a rather simple aim in mind: to make interaction with software easier by making it more intuitive. Educational software has the grander aim of assisting cognition. This aim may have consequences which conflict with requirements of direct manipulation, as evidenced by the black box, control, display and interaction paradoxes. As such direct manipulation may not always be the most appropriate form of interaction for educational software.

REFERENCES

1. Squires, D.J. (1994) *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.* PhD, University of London.

2. Squires, D.J. (ed) (1991) *Bioview.* King's College London, London.

3. Card, S.K., Moran, T.P. and Newell, A. (1983) *The Psychology of Human Computer Interaction.* Lawrence Erlbaum Associates, Hillsdale, New Jersey.

4. Olson, J.R. and Olson, G.M. (1990*) The growth of cognitive modelling in human-computer interaction since GOMS.* Human-Computer Interaction, 5 (2 & 3) pp. 221-265.

5. Gugerty, L. (1993*) The use of analytical models in human-computer-interface design.* International Journal of Man-Machine Studies, **38**, pp. 625-660.

6. Hutchins, E.L., Hollan, J.D. and Norman, D.A. (1986) *Direct Manipulation Interfaces in User Centred System Design: New Perspectives on Human-Computer Interaction* (eds. D.A. Norman and S.W. Draper), Lawrence Erlbaum Associates, Hillsdale, New Jersey, pp. 87-124.

7. du Boulay, B., O'Shea, T. and Monk, J. (1981) *The black box inside the glass box: presenting computing concepts to novices.* International Journal of Man-Machine Studies, **14**, pp. 237-249.

8. Grudin, J. (1989) *The case against user interface consistency.* Communications of the ACM, **32** (10) pp. 1164-1173.

9. Draper, S.W. (1986) *Display Managers as the Basis for User-Machine Communication, in User Centred System Design: New Perspectives on Human-Computer Interaction* (eds. D.A. Norman and S.W. Draper), Lawrence Erlbaum Associates, Hillsdale, New Jersey, pp. 340-352.

10. Norman, D. (1991) *Cognitive Artifacts, in Designing Interaction: Psychology at the Human-Computer Interface* (ed. J.M. Carroll), Cambridge University Press, Cambridge, pp. 17-38.

11. Birnbaum, I. (1990) *The assessment of IT capability.* Journal of Computer Assisted Learning, 6 (1) pp. 88-97.

12. Lucas, A.M. and Tobin, K. (1987*) Problems with "control of variables" as a process skill.* Science Education, **71** (5) pp. 685-690.