In G. Kadoda (Ed). Proc. PPIG 13 Pages 225-235

Salvation for Bricoleurs

Tzippora Yeshno and Mordechai Ben-Ari
Department of Science Teaching
Weizmann Institute of Science
Rehovot 76100 Israel
nttzipiy@wisemail.weizmann.ac.il
moti.ben-ari@weizmann.ac.il

Keywords: POP-I.C. end-user applications, POP-II.A. novice/expert, POP-V.A. mental models

Abstract

Users demonstrate bricolage (aimless trial and error) in their interaction with applications like word processors. This can be attributed to behaviorist and minimalist task-oriented learning. We show that explicitly teaching a conceptual model can improve both the task performance and the understanding of the learners.

Introduction

In a previous presentation at this workshop (Ben-Ari, 1999), the second author described an experiment in which (even) science educators studying for advanced degrees displayed abysmal performance on non-routine tasks using a word processor. The performance was described as *bricolage*, a term first used by anthropologist Claude Levi-Strauss to describe the primitive science of a 'savage', and appropriated by Turkle and Papert (1990) to (appreciatively) describe learning based on trial-and-error rather than planning. While trial-and-error is useful if it leads to refinement of concepts, we reserve the term bricolage (and use it more or less pejoratively) for *aimless* trial-and-error that does not seem to modify concepts or improve the quality of task performance.

The results can be explained within the framework of mental models (Norman, 1983): the traditional methods of learning the use of software packages are not conducive to the development of viable mental models. The primary learning method is the minimalist tutorial that is totally task-oriented. The situation is exacerbated by the fact that such tutorials are frequently given by friends and family who are not experienced educators, or are used for self-study. Even formal teaching tends to concentrate on task performance rather than on understanding.

Is the situation hopeless? Is there no salvation for bricoleurs?

We describe an experiment which shows that the teaching of an *explicit conceptual model* can produce a significant improvement in the quality of learning, as demonstrated both by successful task performance and by student understanding. The construction of an explicit conceptual model is a difficult task which must be done by a qualified teacher or an educator who is experienced in the development of learning materials. However, once the model has been worked out it is not difficult to teach or to learn; on the contrary, a clear, concise conceptual model is both less time-consuming to learn and more efficient than a large set of task-oriented rules.

Previous research

The term conceptual model is taken from Norman:

> *A conceptual model is invented to provide an appropriate representation of the target system, appropriate in the sense of being accurate, consistent, and complete. Conceptual models are invented by teachers [our emphasis], designers, scientists, and engineers.* (Norman, 1983, p.7)

Norman places the conceptual model between the *target system*–the real world or an artifact–and the *mental model* constructed by the user of the target system. While a simplistic approach to constructivism would have learners directly construct a mental model from a target system, Norman envisions a representation intermediate between the two. The teacher is no longer just mediating in the construction of a mental model by questions and negotiation, but is now required to create learning materials that are used by the student in the construction.

Young (1981) performed an experiment on users of calculators. One group received operational instructions and the other received a conceptual model called a *cover story* describing the mechanism of the calculator. The cover-story group worked more efficiently and successfully completed more of the assignments than the other group. Similarly, Kieras and Bovair (1984) presented three studies concerned with learning how to operate a simple control panel, and showed that learning is improved when the learners are given a conceptual model of the internal mechanism of the device. These two studies concern hardware target systems that the users actually see; we are interested in target systems which are software packages—abstractions of real-world artifacts like paper documents.

The second author examined constructivism in the context of computer science education (Ben-Ari, in press) and claimed that the main conclusion that can be reached is that teaching an explicit [conceptual] model is essential. The argument is similar to the one used by Smith, diSessa and Roschelle (1993), who claim that knowledge is constructed on top of misconceptions, where they define misconceptions as prior knowledge that is used outside of the context in which it is viable. If knowledge is constructed as mental models, it must be constructed from something. In the absence of a reasonably viable naive model of a computer or software system, an explicit model should be constructed and presented.

Conceptual models are not limited to artifacts like computers. The idea is similar to the *intermediate causal models* proposed by White (1993) to provide (computer-based visualizations of) artificial models—in fact a hierarchy of such models—to enable the student to bridge the gap between the experiential world and the abstract mathematical description of science embodied in Newton's or Ohm's laws. Using White's terminology, conceptual models are similar to intermediate causal models in that they are also designed to be understandable, learnable and usable. However, they are different in that they are not transferable (decontextualized) nor linkable (linking levels of abstractions).

In learning to use a software system, the sole goal is viable task performance. There is absolutely no expectation that an average user achieve *understanding* of a word processor! Furthermore, models are likely to be very context-dependent and non-transferable (except within a family of software systems). There are two further specific premises of this work:

- The computer artifact is given and must be taught. Norman concluded that a stack calculator, once learned, is easier to use than a four-function calculator. However, what is an educator to do if the use of a four-function calculator is mandated, perhaps by the school administration or by company policy? Similarly, there are word processors that, once learned, are easier to use than MS Word, but given the dominant position of MS Word, its use is likely to be mandated, and the educator must accept the challenge or be irrelevant.
- There is no doubt that professional designers of any software system are guided by a conceptual model (cf. the quote from Norman above). But there is also no doubt that such models are rarely published, and even if they were, would likely be too complex for the average user to master.

Problem statement

Microsoft Word is one of the most widely used software packages. Word is highly sophisticated, capable of performing the most advanced formatting of documents. However, the sheer number of menu selections, icons and options can make it overwhelming for a novice. As shown in Ben-Ari (1999), even first-rate science educators employ bricolage in their attempts to create documents with

Word. Several problem areas were identified such as the inclusion of pictures and tables, and the formatting of lists.

In the research described here, we chose to focus on one aspect of the use of Word: entry and modification of dual-language text in the Hebrew language version of the software. This choice is justified since the feature is quite complex (so that the measurement of a treatment effect will be easy), yet it is in constant use by users of all levels (so that students will be motivated to learn).

The difficulty with Hebrew text entry is not in the use of a different alphabet, but in writing the text from right to left (RTL). That in itself would not be a problem were it not for the fact that numbers and arithmetic expressions are written in left to right (LTR) order. Furthermore, in technical writing, terms such keywords of a programming language are also written from LTR. To accommodate these requirements, Word supplies commands that can be invoked either from icons on a toolbar or from the keyboard (e.g. left-alt+left-shift toggles between LTR and RTL mode). The current language is indicated by a small notation on a toolbar and by the shape of the cursor.

To simplify user interaction, changes in either language or direction are often performed automatically (e.g. when writing in English in a RTL text, pressing the End key will change (back) to Hebrew). It is precisely this user-friendliness which is at the root of users' difficulties: without a good mental model of the algorithms used by Word, it is impossible to explain or predict the result of any key press. For example, suppose that you write the following sentence about a programming language (where Hebrew letters have been replaced by lower English letters from the end of the alphabet):

yyyyyyyy **if** **then** xxxxxxx

and you decide to write the Hebrew word zzzz for condition between the **if** and the **then**. The result will be:

yyyyyyyy **then** zzzz **if** xxxxxxx

In Norman's terminology, the target system is behaving in a perfectly logical and consistent manner, and furthermore, the designers of this feature had a clear conceptual model, though this is not published for users to study. The end user who must build a mental model directly from the behavior of the target system is very likely to be mystified at the exchange of the English words!

Research question

We wish to investigate the performance and understanding of students dealing with complex tasks in an application like MS Word. The treatment group will be taught using an explicit conceptual model and the control group will be taught using conventional methods. We hypothesize that the treatment group will demonstrate higher levels of performance and understanding, as well as reduced anxiety.

Methodology

Preparatory steps

The first step in the research was to fully understand the target system—not a simple task! (For reference, let us note that we are describing version 97 of MS Word.) The tutorials and help document were task-oriented and did not contribute to understanding how Word works. The next step was to create a very technical conceptual model which was expressed in a state transition diagram. A state was defined as a quadruple:

<direction><language><cursor-style><cursor-position>

where ¶◄ and ►¶ are the symbols used by Word to denote the direction, <En> and <He> denote the language and there are three cursor styles [,], |, for typing English characters, Hebrew characters, and numbers and special characters. The cursor position refers to the position of the cursor relative to the last character entered. State transitions occur when pressing keys in one of the categories: Hebrew character, English character, digit, symbol, Home key and End key. State diagrams were created for four tasks: (a) creating a new document, (b) inserting characters into a document, (c) erasing characters from a document, (d) changing characters in a document.

It should be clear even from this sketch that the state diagrams were exceedingly complex and useless as a conceptual model from which to learn. The next step was to elaborate a simpler conceptual model of how Word works based on the concept of *blocks*. A block is a sequence of characters typed in one language. In a transition from one language to another the old block closed and a new block is opened. The direction of the cursor while moving in the block depends on the language and the type of the characters that the block contains. In a block written in English, the direction of the cursor movement is from left to right. In a block written in Hebrew, the direction of the cursor movement is from right to left, except for digits and the special characters +, -, %, /.

The conceptual model was used as the basis for a tutorial which analyzes the behavior of the word processor in the two editing directions. The emphasis in the tutorial is in presenting the conceptual model and then with giving conceptual, rather than task-oriented, explanations for problems that the user is likely to encounter.

Preliminary experiment

The tutorial was first evaluated by giving it to two high-school teachers, one a teacher of computer science and the other a teacher of business. The first teacher often uses Word; the other teacher teaches the use of Word. After reading the tutorial, the teachers were presented with several exercises concerning typical problems that occur while editing text that requires the inclusion of English material such as numbers within a Hebrew document. The teachers were observed while they performed the exercises.

The first exercise was to type a date preceded by a hyphen:

17/12/00-xx

Both teachers recognized that there is a problem because the hyphen character is interpreted as an arithmetic minus sign and thus would be written from left to right when trailed by digits:

-17/12/00xx

They both knew that they had to change languages but they could not figure out exactly *when* to change languages until after block model was explained. Before that a typical response was: “I’m sure that I have to change language but it doesn’t work…. I still get the sign from left to right.” In another exercise they could not find the right place to position the cursor before typing. The exercise is to place Hebrew letters yy between the hyphen and the 5 in the string:

12345-xxxx

To succeed, you must place the cursor to the left of the 1, because that is the start of the numeric block! A typical response was: “I don’t know how to do it. I’m sure that the position of the cursor is right but when I insert the new character all the text gets mixed up.”

From this experiment, the importance of the block model to understanding the direction and the order of the characters became clear. The tutorial must explicitly use the concept of working within the block model to explain the transition from one language to another. Without such a model to organize the users' view of the software, they regress to bricolage which leads to solution only by chance.

Main experiment

In the next stage, the first author taught the material to three groups of ninth-grade pupils, a control group of 17 pupils and two treatment groups of 19 and 13 pupils. A questionnaire was used to elicit general information about computing experience; analysis showed that the groups were very similar. However, the first treatment group started its study of Word later in the school year.

In all the groups, the lesson began with a discussion of the problems of dual-language word processing This discussion served as a starting point for an explanation of how Word deals with these problems. The control group received explanations as operational procedures: for each problem the solution was given. In the experimental groups, the solutions were explained in terms of the block conceptual model. Both groups then received a worksheet for self-study, followed by an examination. A full description of the worksheets and examination will appear in the final report of this research.

Results

Worksheets

This section presents the results obtained by analyzing the worksheets. Five categories were defined representing progressively higher levels of understanding:

Category 1 - They successfully complete simple assignments concerning the cursor style, direction of the cursor, and position and style of cursor when pressing the End or Home key.

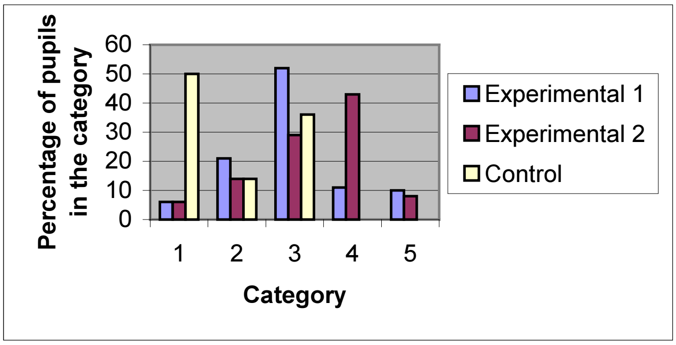
Category 2 - They understand what happens to text when entering characters in one language between characters in the other language. This understanding holds for both paragraph directions.

Category 3 - As in the second category and in addition they partially understand how to maintain the correct direction while dealing with special characters and digits.

Category 4 - As in the second category and in addition they completely understand how to maintain the correct direction while dealing with special characters and digits.

Category 5 – They understand the block model.

Here is the distribution of pupils in the categories:



We see that the control group is very good in simple assignments, but that none of them succeed in solving the problem of keeping the correct direction while entering digits between other characters (the fourth category). (Note that the last category is not relevant to the control group.) All the pupils tried to transfer from one language to another, but only some of the pupils from the experiment groups understood what to do and how to do it. These pupils described the solution in terms of the block model by sketching the blocks of each group of characters.

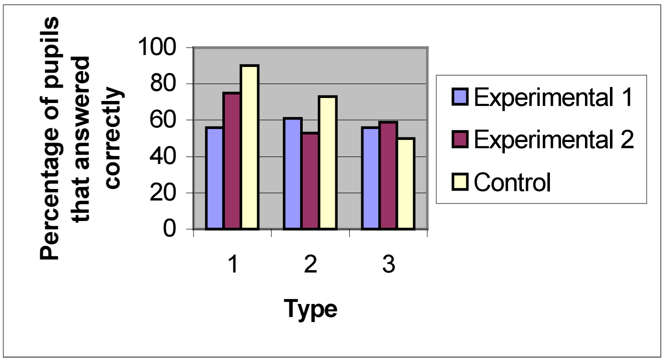
Examinations

Following the worksheet, a multiple-choice examination was given, which checked task-oriented knowledge. There were three types of questions:

1. Questions that checked the pupils' knowledge of the procedures used to change from one language to the other and from one paragraph direction to the other.

2. Questions that checked the pupils' knowledge in both paragraph directions when entering characters in one language between characters of other.
3. Questions that checked the pupils' knowledge about cursor styles and directions in both languages.

To solve questions of the first two types, procedural knowledge is sufficient; for the third type, conceptual knowledge is needed. The distribution of the correct answers is given in the following chart:



Discussion

Worksheets

From the analysis of the assignments, we found that pupils from the experimental groups can employ the conceptual model of blocks to verbalize their answers (orally and in writing); this also leads to improved performance on tasks. The control group pupils dealt with the assignments in different manner, recalling bricolage, and could not verbalize their attempts to solve the assignment.

Examinations

Unfortunately, the control group took the examination under conditions that allowed copying. This accounts for the high scores achieved in questions of the first two types. However, even given the possibility of copying, they scored lower than the experimental groups on questions of the third type.

A teacher's reaction

The reaction of the teacher of both the second experimental group and the control group was quite interesting. She has been teaching Word for three years in high school and has participated in several training courses. She observed while the first author taught the conceptual tutorial, and at the end of the class claimed that even for her, the tutorial helped her deal with problems that she had had. The block model gave her tools for systematically solving problems. While this research targeted novice learners, further research could target the effect of conceptual models on users with extensive, task-oriented knowledge.

Conclusion

The results from these experiments are encouraging and support our claim that explicitly teaching a conceptual model provides tools for systematic and insightful work, leading to better problem-solving performance. We intend to repeat the experiment in order to better control the experimental conditions and to perform statistical analysis.

Learning to use a software application is perceived to be a difficult and abstract task. There is a need to create instructional approaches that make software applications less intimidating and accessible to a wide range of users. The approach described here builds on the cognitive science concept of a mental model. We believe that *teachers* need to create conceptual models and to explicitly teach them to novice users. The conceptual models will enable the users to develop viable mental models of applications, increasing their successful performance of tasks and decreasing the anxiety that they feel.

Acknowledgements

We would like to thank Abraham Arcavi for his suggestions during this research and for his helpful comments on the paper. We would also like to thank the referees for their comments.

References

Ben-Ari, M. (1999) Bricolage Forever! *PPIG-11 Workshop,* 53-57.

Ben-Ari, M. (in press) Constructivism in computer science education. *Journal of Computers in Mathematics and Science Teaching*. Preliminary version in: *SIGCSE Bulletin* 30(1), 257-261, 1998.

Kieras, D.E., Bovair, S. (1984) The role of a mental model in learning to operate a device. *Cognitive Science*, 8, 255-273.

Normal, D.A. (1983) Some observations on mental models. In: D. Gentner and A.L. Stevens (Eds.) *Mental Models,* 7-14, Hillsdale NJ: Lawrence Erlbaum Associates.

Smith III, J.P., diSessa, A.A., Roschelle, J. (1993) Misconceptions reconceived: A constructivist analysis of knowledge in transition. *The Journal of the Learning Sciences*, 3(2), 115-163.

Turkle, S., Papert, S. (1990) Epistemological pluralism: Styles and cultures within the computer culture. *Signs: Journal of Women in Culture and Society*, 16(1), 128-148.

White, B. (1993) Intermediate causal models: A missing link for successful science education? In: R. Glaser(Ed.) *Advances in Instructional Psychology*, 4, 177-252, Hillsdale NJ: Lawrence Erlbaum Associates.

Young, R.M. (1981) The machine inside the machine: user's models of pocket calculators. *International Journal of Man-Machine Studies*, 15, 51--85.