https://doi.org/10.48009/2_iis_2010_191-198

NEARLY WORDLESS TECHNIQUES FOR SOFTWARE INSTRUCTION: INCREASING COMPREHENSION AND HIGHER LEVEL LEARNING

Raymond D. Frost, Ohio University, frostR@ohio.edu
Sarah Pels, Ohio University, sarah.pels@gmail.com

ABSTRACT

This paper describes the creation of nearly wordless techniques to aid in software instruction. The techniques depict procedural information visually and conceptual information verbally. The techniques are designed to promote critical thinking and higher order learning while teaching software skills. The authors provide background information about the techniques, their relation to learning theory, and the course in which they are used. Principles from allied fields are adapted to develop a lexicon for software instruction. The paper develops the lexicon and describes the nearly wordless diagrams spawned from it. Implications and future areas of research are discussed.

Keywords: Techniques for software instruction, instructional design, nearly wordless lexicon, introductory course, critical thinking.

INTRODUCTION

The authors' goal was to increase comprehension and higher level learning in an introductory management information systems course taught both in class and online. This paper examines one component of the instructional design for that course. Particular attention is given to the use of nearly wordless diagrams to promote effective software instruction. The assumption behind this research is that if students can master basic software concepts quickly then they will have more time and energy to devote to business and design goals.

The software used in the course includes Microsoft Word, Excel, PowerPoint, and Google Sites. The theory taught in the course includes business process, interface design, market segmentation, the systems development life cycle, budgeting, performance measures, reporting, presentations, analytical design and graphic design.

There are two issues in software instruction texts—what is taught and how it is taught. Faculty often express more frustration with the "what" issue rather than the "how" issue. For example, faculty might complain that a concepts book in the introductory course contains a fairly high level discussion on business intelligence, but is then followed up with a fairly trivial Microsoft Access lab exercise. Neither students nor faculty readily see any connection between the two.

We would argue that one reason many exercises are trivial has to do with how the material is taught. Most software instruction takes the step-by-step and screen-by-screen approach. Each keystroke is meticulously described. The problem with the step-by-step approach is that students learn keystrokes but not concepts: students do not know when to apply certain skills, nor how to employ those skills to support business or design goals. In sum, the exercise is doomed to be trivial.

But why do instructors teach this way? After all students master video games with little or no instruction. They do not plow through step by step guides to learn the video game. They experiment continuously, and if they get stuck they might seek out a hint. Yet when it comes to software instruction instructors feel compelled to put students in a keystroke straightjacket. Why? More time should be spent describing and selling the business purpose and less time describing exactly how to get there.

Step by step instructions have their proper scope and proper expression. Instructions should demonstrate particular techniques, not to walk a student through the entire exercise. The verbal dimension should be reserved for critical reflection about how and when to apply the technique. The proper expression of step by step instructions should be visual rather than verbal. Students want to see how to complete the technique rather than just follow a long list of steps. The visual preference may well be responsible for the popularity of "how to" videos on YouTube. And while videos are useful, they do not serve well as a quick reference guide.

LITERATURE REVIEW

Changes in instructional design have a greater chance of success when based in accepted learning theory. One of the most recognizable contributors to learning theory is Bloom who created a taxonomy that proceeds from rote memorization at its lowest level to evaluation of alternatives at its highest [2]. Starting at the lowest level, the six levels of the taxonomy are: knowledge, comprehension, application, analysis, synthesis, and evaluation. The techniques under discussion are designed to support the comprehension and application levels of the taxonomy.

In order to achieve these levels of learning, a GOMS model is employed. The Goals, Operators, Methods, and Selections (GOMS) model [4] is a system for describing task performance. Using the GOMS model, each assignment is broken down into a hierarchy of goals. The lower level goals are referred to as steps, which are grouped together to form a method or technique. It is the responsibility of the instructional designer to make decisions about the task. The decisions include determining which series of steps form a method and which steps can be omitted due to background knowledge of the student.

There are multiple ways to depict motion in instructional diagrams. De Souza and Dyson identify four categories: composite images, synoptic images, before-during-after images, and significant moments. Of particular interest to the current research are significant moments and synoptic images. Significant moments, or single images, represent snap shots in time. It is very difficult to determine motion from significant moments. A synoptic image is set of moments compiled into a single unit. Synoptic images have four advantages [5]. Synoptic images:

- "portray the action as a whole
- depict motion displacement more directly
- inform trajectory more directly
- save space"

One of the most prevalent uses of synoptic images is in product guides. Product guides have a strong motivation to go nearly wordless as some of them must otherwise be translated in to up to nineteen different languages. When done well, product guides also have the ability to convey a wealth of information with few images and few words.

Rodriguez develops a set of design principles for troubleshooting laser printers and then tests the consequences of breaking key guidelines [15]. Martin et al. test the usability of toy assembly instructions and compile a set of guidelines [11]. In an extensive series of experiments, Agrawala et al. inductively develop a series of design principles for furniture assembly. They show how instructions based on those principles can be computer-generated [1].

A consistent theme in all of the literature is that it is the *quality* of the instructions that is paramount [1, 17, 22]. Quality is determined by two factors. The first factor is to operate from design principles. The second factor is to consistently follow those principles [1, 15].

The researchers cited above agreed on a number of principles. These similarities are reflected in Table 1 below.

According to Wandersee et al. an effective diagram serves as an initial crutch for students in the early stages of learning [13]. It is progressively less effective as the student masters the subject. The task is to improve the diagrams and make them more effective.

METHODOLOGY

Each unit in the course requires three assignments designed to support the theory discussed in that unit. The assignments proceed in three levels labeled L1, L2, and L3 which are progressively challenging. The entire course is built around students designing an original iPhone app. For example, at the L1 level a student provides the demographic profile for market segments that are interested in the app. The L2 assignment requires students to diagram the As-Is and To-Be process flows for the service that the iPhone app provides. The L3 level assignment requires students to design and mockup multiple screens from the proposed app.

The L1 and L2 assignments are designed to focus on the comprehension level of Bloom's taxonomy. The L3 assignments target the application level.

The course is challenging in that students must master of a number of software programs in a one-term course. In order to utilize the software in an original setting, students must understand the skills very well. Therefore, the challenge is how to present the skills in a clear, concise and memorable way without actually demonstrating how to do the assignment.

The authors determined which tasks were required to complete each assignment. A instructional technique, the nearly wordless diagram, was developed for each task. After much experimentation the diagrams were composed using synoptic images in order to better depict motion. Thus procedural information is conveyed with images while conceptual information is conveyed in writing.

The techniques utilize an explanatory framework developed by Richard Paul to promote critical thinking. Paul encourages faculty to communicate concepts in four forms: 1) provide a definition, 2) rephrase the definition "in other words," 3) provide a written example, and 4) graphically illustrate that example. The first three forms in the framework are provided in a caption to the diagram. The fourth form, the graphical illustration, is the diagram itself.

Building on the previously cited literature, each design rule from Table 1 was applied to the field of software instruction. The principles are defined below and examples showing their use appear at the end of this paper.

The design rules serve two functions. First, they are used to establish a lexicon. A lexicon typically refers to the words and expressions of a language. In nearly wordless diagrams, the "words and expressions" are the symbolic representations of a combination of mouse and keyboard actions. The lexicon provides a standard language for use in the diagrams. Figure 3 below reproduces part of the lexicon.

Second, the design rules contribute to the layout and structure of the diagrams. For example, the design rules dictate where and at what size each screen element should appear. By applying these design principles along with the GOMS model, ninety nearly wordless techniques were created. An annotated version of one of the techniques is shown in Figure 1 and contrasted with the type of technique it replaces in Figure 2.

1. **Parsing rule**—Techniques are divided into logical units of about three to four steps each. For example, there are Excel techniques to calculate future value, sort data, and autosum. The sort data technique shows the following steps: highlight the data, select the sort data icon, and then select sort options.
2. **Object of focus rule**—Most graphical user interfaces naturally highlight one object of focus, for example causing a menu to glow on mouse-over. However, as the user completes each step, the object of focus changes. The challenge in a single diagram is to establish an initial object of focus and then move that object of focus around the "screen" for each step. For example, to create the sort data technique required superimposing portions of multiple screen shots to create multiple objects of focus. This, combined with the grasping point and sequence rules below, effectively move the object of focus around the screen.
3. **Location rule**—Screen elements are all oriented relative to each other as they would appear to the user. For example, in PowerPoint, the outline view appears on the left, and the menus at the top.
4. **Grasping point rule**—Onscreen objects are "grasped" with a mouse click and drag. A starburst, mouse-pointer, and arrow represent the click and drag event.
5. **Action rule**—A lexicon was created including various click-events, keystrokes, and state-changes of the mouse-pointer. For example, in PowerPoint the mouse-pointer changes state when positioned over the green handle of a bounding box to enable the rotation of an image.
6. **Motion rule**—The lexicon also includes conventions for mouse-movement in combination with or without click events. Solid or dashed arrows are used to depict mouse-movement.
7. **Sequence rule**—The lexicon numbers steps in logical order.
8. **Feedback rule**—The diagrams present visual feedback as to how the object will look and behave following the action.
9. **Arrangement rule**—Where possible steps are arranged top to bottom and left to right. However, in cases where the arrangement rule conflicted with the location rule, the location rule received a higher priority.
10. **Picture size rule**—Pieces of individual screen shots were superimposed so that each menu and icon could be enlarged to a readable and user-friendly size.

CONCLUSIONS AND FUTURE RESEARCH

The goal is to improve the software deliverables in an introductory management information systems course. To reach that goal requires empowering students to produce professional work products.

Anecdotally, the results are outstanding. Comprehension and retention have been improved. Students have fewer "how-to" questions. In addition the work products of the students are more sophisticated and demonstrate greater creativity. It seems as though students have more time and energy to focus on the higher order learning objectives.

Future research could empirically test the efficacy of the nearly wordless diagrams. Such research could look for the ideal balance of words and graphics in the diagrams. Another possible research stream could focus on whether the nearly wordless diagrams better transcend language boundaries in a global learning environment.

REFERENCES

1. Agrawala, M., Hanrahan, P., Haymaker, J., Heiser, J., Klingner, J., Phan, D., & Tversky, B. (2003). Designing Effective Step-By-Step Assembly Instructions. *ACM Transactions on Graphics*, 828-837.
2. Bloom, B. S., et al. (1956). *Taxonomy of Educational Objectives: Handbook I Cognitive Domain*. New York: David McKay.
3. Brasseur, L. E. (2003). *Visualizing technical information: a cultural critique*. Amityville, N.Y: Baywood Pub.
4. Card, S., Moran, T., & Newell, A. (1983). *The Psychology of Human-Computer Interaction.* Hillsdale, NJ: Lawrence Erlbaum Associates.
5. Dyson, M., & D'Souza, J. (2008). An illustrated review of how motion is represented in static instructional graphics. *STC's 55th Conference.*
6. Dyson, M., & D'Souza, J. (2008). Are animated demonstrations the clearest and most comfortable way to communicate on-screen instructions? *Information Design Journal*, *16*(2), 107–124.
7. Fleming, N. D. (1995). I'm different; not dumb. Modes of presentation (VARK) in the tertiary classroom. *Research and Development in Higher Education.* Proceedings of the 1995 Annual Conference of the Higher Education and Research Development Society of Australasia (HERDSA). 18 308 – 313.
8. Holmes, N. (2005). *Wordless Diagrams*. New York, NY: Bloomsbury Publishing.
9. Conrad, N. K., Dragich, D., Jalongo, M. R., & Zhang, A. (2002). Using Wordless Picture Books to Support Emergent Literacy. *Early Childhood Education Journal 29*(3), 167-77.
10. Krull, R., Roy, D., Sharp, M. D., & D'Souza, S. (2004). Designing Procedural Illustrations. *IEEE Transactions on Professional Communication*, *47*(1), 27-33.
11. Martin, C. V., & Smith-Jackson, T. L. (2008). Evaluation of pictorial assembly instructions for young children. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *50*(4), 652-662.
12. Mergal, B. (1998). Instructional Design and Learning Styles. Available: http://www.usask.ca/education/coursework/802papers/mergel/brenda.htm
13. Mintzes, J. J., Novak, J. D., & Wandersee, J. H. (1998). *Teaching Science for Understanding*. San Diego, California: Academic Press.
14. Morse, D. L., & Novick, L. R. (2000). Folding a fish, making a mushroom: The role of diagrams in executing assembly procedures. *Memory & Cognition*, 28, 1242-1256.
15. Poison, P. G., & Rodriguez, M. A. (2004). Creating usable wordless instructions for performing complex one-time tasks: Effects of violating the rules. *Proceedings of the Human Factors and Ergonomics Society 48th Annual Meeting*, 941-945. Santa Monica, CA: Human Factors and Ergonomics Society.
16. Reynolds, G. (2008). *Presentation Zen*. Berkeley, CA: New Riders.
17. Rodriguez, M. A. (2002). Development of Diagrammatic Procedural Instructions for Performing Complex One-Time Tasks. *International Journal of Human-Computer Interaction, 14*(3), 405-422.
18. Szlichcinski, C. (1984). Factors affecting the comprehension of pictographic instructions. In R. Easterby & H. J. G. Zwaga (Eds.), *Information design*: 449-466. Chichesler, UK; Wiley
19. Tufte, E. (1990). *Envisioning Information*. Cheshire, CT: Graphics Press.
20. Tufte, E. (1997). *Visual Explanations*. Cheshire, CT: Graphics Press.
21. Tufte, E. (2006). *Beautiful Evidence*. Cheshire, CT: Graphics Press.
22. Wright, P. (1982). Manual dexterity-a user-oriented approach to creating computer documentation. *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*.

	Rodriguez	Agrawala	Martin
1. Parsing rule **Divide action into logical and separate tasks**	X		
2. Object of focus rule **User should be able to clearly identify which part(s) is being acted on in each step**	X		X
3. Location rule **It is important to show orientations of the object in a manner that is physically realizable.**	X		X
4. Grasping point rule **Where something is being acted on should be unambiguous**	X		
5. Action rule **How something is being acted on should be unambiguous**	X		
6. Motion rule **Use consistent use of diagrammatic elements to show motion**	X		
7. Sequence rule **Use clear and explicit order for performing actions**	X	X	
8. Feedback rule **Show the expected end-state of each action**		X	
9. Arrangement rule **Arrange from left to right and top to bottom**			X
10. Picture size rule **Pictures should be sufficient to discern parts by color, shape, size**			X

Table 1. Design Rules. Rule titles 1-9 taken from Rodriguez (2003). Rule descriptions 1-9 taken from Agrawala (2003) Rule and description 10 taken from Martin et al. (2008)

Figure 1. Nearly Wordless Technique: Focus on Design and Critical Thinking.

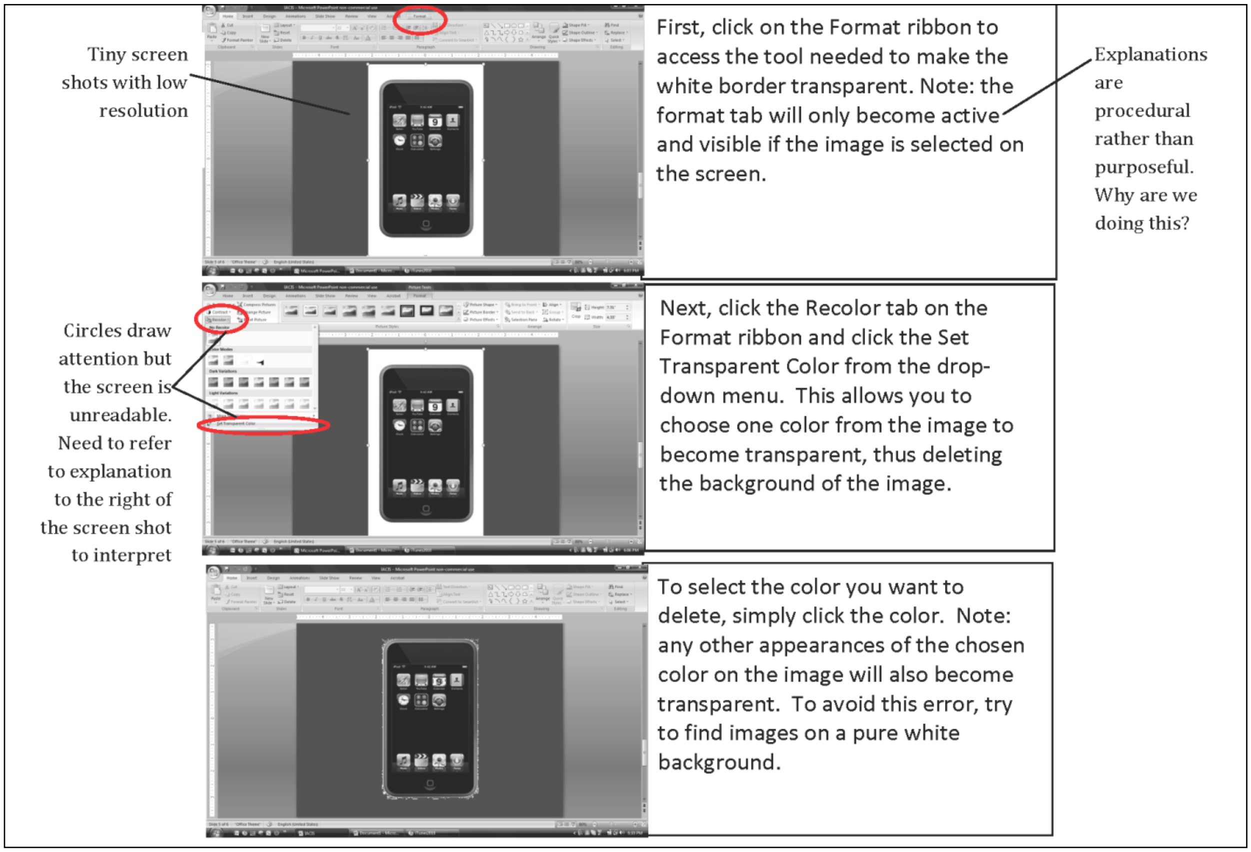

Figure 2. Step by Step and Screen by Screen Technique: Requires Words to Describe Action

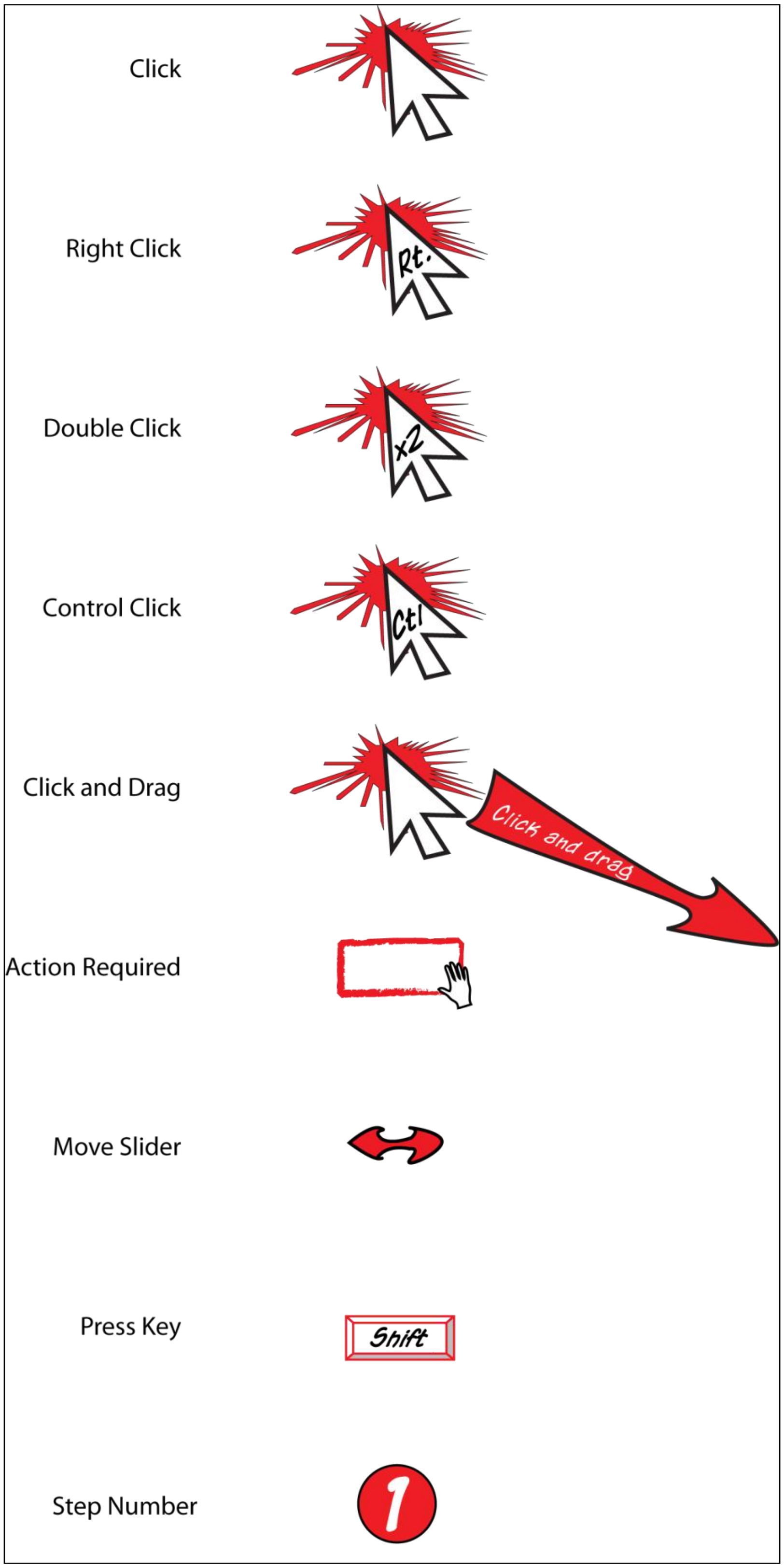

Figure 3. Lexicon for Nearly Wordless Diagrams