Constantine & Lockwood, Ltd.

Studies in Applied Usage-Centered Design

Design Study 1: Active Table-of-Contents Control for Content Navigation and Customization

Larry L. Constantine
Lucy A. D. Lockwood
Constantine & Lockwood, Ltd.
http://www.forUse.com

Abstract: *This report illustrates the design of a novel user interface feature to provide simple and rapid navigation and user customization of the contents of a complex, multipart document. Within a performance-support application for classroom teachers, the objective was to provide an efficient and instantly learnable scheme for direct user control over the parts to be included in the document as well as quick access to any part of the document. The design relies on the techniques of instructive interaction, an innovative approach for making user interfaces self-teaching even when they incorporate novel or non-standard features.*

Keywords: document navigation, customization, user control, instructive interaction, usage-centered design, design innovation, user interface design, interaction design, usability, performance support

Series Introduction

Often times we have been asked to give examples of our work or that of other designers practicing usage-centered design [Constantine and Lockwood, 1999]. In our own teaching and consulting, we typically emphasize the process by which design solutions are developed and de-emphasize the solutions themselves. In our view, it is more important for designers to learn how to design for use—how to find effective answers to their own problems—than to see the answers that others have devised as solutions to other problems. However, in many cases the best way to illustrate a process is through a concrete example from the real world, especially as carried out by particularly experienced practitioners.

With the appearance of this white paper, we begin a new series of reports based on actual design problems solved through application of usage-centered design. In each of these design studies, we will feature a discrete problem from our own work for and in collaboration with clients. Each study not only will illustrate the course of problem solving and the evolution of a design but also will present an

The authors may be contacted at Constantine & Lockwood, Ltd., 58 Kathleen Circle, Rowley, MA 01969; tel: 1 (978) 948 5012; fax: 1 (978) 948 5036; mailto:larry@foruse.com | mailto:lucy@foruse.com

instance of the kind of innovative solution we often seek in our own design work. In effect, we hope to give readers what amounts to a glimpse over the shoulders of experienced visual and interaction designers at work. We have selected these particular examples because they represent interesting challenges that are far from unique and because the solutions we devised may indeed have broader application in other contexts.

The first several studies in this series are all derived from a single project: a classroom information management system for K-12 classroom teachers. The technical, business, and user requirements of this project required a groundbreaking user interface based on novel and more effective solutions to a variety of visual and interaction design problems. We are pleased now to be able to share some of this work with our colleagues.

General Background

We were retained by an Internet startup to help design a browser-resident classroom information management system for K-12 classroom teachers. This sophisticated application was a form of performance-support system intended to empower classroom teachers by facilitating and simplifying key administrative, planning, and teaching tasks in an environment fully integrated with software and Web-based learning resources and existing administrative systems. The application was to be deployed through a Web-based technology with the client-side user interface realized in HTML, XML, and Java JFC/Swing.

A key facility of the system was to be its support for lesson planning. Classroom teachers in contemporary settings are responsible for preparing in advance detailed, written lesson plans for all material they teach.

Lesson plans specify such things as the objectives of each lesson, its subject matter, the materials to be used, the methods of instruction employed, and the criteria and means for evaluation of student learning. In many schools, individual lessons must also be tied to the related state or district-defined standards specifying what must be achieved in each subject at each grade level, in this way documenting the performance of the school and teacher and their conformance to mandated standards.

Lesson plans as paper documents take many forms, and there is great variability among the contents and formats used, depending on the grade level, subject, school or district, and individual teachers. Nevertheless, there are many common elements in lesson plans and some contents, such as title and objectives, may be considered required content in all cases. For some purposes, certain conventional formats of presentation have been adopted as more or less standard. Even so, individual teachers or groups may prefer their own custom selection of what categories and kinds of information are to be included.

Typical classroom teachers may have less than 15 minutes a day to plan the entire day's lessons. Often, the starting point is an already existing lesson plan that is reviewed and modified, which could involve customizing a plan obtained from elsewhere not only by changing the content but also by incorporating additional sections or categories of description while omitting others. In some cases, the teacher may need to create a new lesson plan from scratch, starting with a blank template or form of some kind and pulling together resources and information from a variety of

places. All this must be accomplished quickly and under conditions where the user could at any time be distracted or interrupted.

Design Objectives

In light of the demands on classroom teachers, we set some rather stringent objectives for the design of the information management system in general and the lesson planning feature in particular. Ready acceptance by classroom teachers was recognized as a key factor in the success of the system.

Ease of learning – It quickly became clear that classroom teachers had little time for learning how to use a new piece of software. We wanted every aspect of the design to be either immediately intuitable or readily mastered with minimal instruction. We defined our objective in terms of a concept we termed the "one-page tutorial." A tutorial of no more than a page of text and illustrations had to be sufficient to enable the average user to make full and immediate use of all the essential features of the system.

Efficiency in use – Although technology experts like to tout the productivity gains possible with the use of computers, historically, many applications actually prove to take more time and to be clumsier to use than their paper-based alternatives. With so little time for such activities as lesson planning, classroom teachers needed a system that was at least as fast and easy to use as the paper-and-pencil forms that they were already accustomed to employing for the same task. Thus, the interface needed to be organized for rapid completion of common tasks and every aspect of the operation of the system needed to be as simple and speedy as possible.

Flexibility – Because classroom teachers work in many different settings and often have developed strong personal preferences for particular ways of working with particular kinds of information, the system should not impose a single way of working or a single organization for information. Wherever possible, it should allow ready and immediate customization to conform to the individual user's ways of working and thinking. In the interest of efficiency and simplicity, however, the user should not have to complete elaborate setup or separate customization processes for simple variations on standard tasks.

Instructive Interaction

The overall goal of achieving an innovative, world-class design that could be put to immediate use by the average classroom teacher required the use of instructive interaction in the user interface. Instructive interaction is an approach to user interface design that we devised to make new systems, including ones based on novel or unconventional components, self-teaching. Through instructive interaction, users learn how to use a system by using it. Instruction is not separated from use. Help, guidance, and assistance are not distinct or separate parts of the system, but are intrinsic to the user interface itself. We developed instructive interaction by drawing on and integrating a diverse body of techniques, both long established and novel. Through the structure, appearance, and behavior of the user interface taken as a whole and in its fine details, the user can be subtly and implicitly guided in the correct use of the system and its elements. A full discussion of the techniques of instructive interaction are beyond the scope of this report, but some examples are illustrated in the design study presented here.

The Problem

Within the classroom application, a lesson plan is a complex and potentially lengthy document organized into numerous sections covering the various component parts of the plan. These include both required contents that are part of every lesson plan—including lesson title, teacher name, grade level, subject, objectives, and activities—as well as optional information or sections—such as, unit and topic, duration, overview, correlated mandated standards, homework, materials needed, resources used, teacher preparation required, assessment, out-of-class extensions, variations for special needs, learning styles involved, teaching styles employed, notes or comments, and other attachments.

To facilitate teacher acceptance and learning, lesson plans within the application need to appear familiar to teachers, although it was recognized that they could not merely slavishly emulate paper-based equivalents. A variety of standardized formats would need be to supported for various school systems, subject areas, and individual teacher preferences. Within differing formats, the required and optional parts may vary as well as the order and arrangement of these as they appear on-screen and in printed copies.

In generic terms, the user interface needs to support the following task cases, among others, in the simplest and most efficient manner:

reviewing contents of a specific section

reviewing contents of the next/previous section

entering/changing contents of a section

excluding a section

including a section

The usability objectives reviewed in the previous section translate into specific goals for the design of the scheme to be provided for navigating and controlling the contents of a lesson plan.

- The emphasis on efficiency requires a fast, simple means of getting to any part of the document from any other. It also favors having as much of the lesson plan visible as possible, since scanning by eye is by far the fastest and most natural means of review and search. However, because the application resides within a browser instance, screen real estate is a particularly scarce resource that must be carefully managed.
- The need for flexibility means that teachers should be able to decide what is included and omitted from a specific lesson plan and should be able to easily review, enter, or modify those parts of the lesson plan that are of interest in whatever order makes sense or is preferred by the teacher, independent of the arrangement of parts within the document.
- The need for ease of learning means that teachers should be able to see how to accomplish these basic tasks immediately upon first seeing a lesson plan as realized within the application.

Design Process

It is important to note that a naïve design that simply realizes a lesson plan as a long, scrollable page filled with simple text entry boxes or fields does not satisfy the

usability objectives in supporting the needed tasks, particularly when it comes to efficiency. If, as is common, only a few scattered sections are of interest to the teacher in reviewing and updating a particular lesson plan, scrolling and scrolling is an inefficient and annoying solution.

The pressure of what ultimately proved unrealistic delivery schedules on this project led us to employ a streamlined version of usage-centered design that took certain shortcuts and derived user interface paper prototypes directly from task cases [see Constantine, 2000; 2001]. Within the background outlined and the given objectives and constraints, our thinking process for evolving a design went approximately as follows.

At first glance, it may appear that there are two distinct problems to be solved here, although they are clearly interrelated. One problem is that of allowing user control over document content, the other is user navigation of the document content. For discussion purposes, let's refer to these two parts of the common design problem by calling them the content selection problem and the content navigation problem.

Content selection

Considering first the issue of content selection, that is, document content customization, we rejected using a separate dialog or screen for this purpose for two reasons. For one thing, this ordinary solution is awkward and inefficient, particularly if customization through inclusion/exclusion of particular sections is a common part of the preparation of individual lesson plans. For another, many users, particularly beginners, are unlikely to use a separate customization facility. Indeed, many users may not even discover the existence of such facilities until after accumulating substantial experience with a system.

In the interest of efficiency as well as accuracy, the user needs to be able to see at a glance just what sections or content are contained in a given lesson plan and what have been omitted. The user should not have to waste time scrolling through a document looking for a section that has, in fact, been omitted from a particular lesson plan, but neither should they have to waste time skipping over blank sections that are not part of that plan. In other words, the user needs to know up front about what sections have been excluded but should not actually see these in the body of the lesson plan.

One fairly obvious possibility is to control the inclusion/exclusion of sections with a series of check boxes. However, a big group of check boxes uses a significant amount of scarce screen real estate. Moreover, a facility for including/excluding particular sections is most likely to be needed only when starting with a lesson plan. However, a modal solution that confronts the user with the content control at the start before allowing review or modification of a lesson plan is unacceptably rigid. Furthermore, a teacher might reasonably decide at some later point in preparation of a lesson plan to include another section or to drop something.

Basically, we want to have a content selection control that hides itself after it is used. A drop-down extended selection list (a drop-down with a series of checkable items) might be used, but the standard behavior of such components is awkward and inefficient in practice. (The drop-down closes after each selection or deselection.) We need a control that works something like the standard one but allows for efficient multiple selections/deselections.

Content navigation

On the other part of the problem, document content navigation, the user needs to be able to go directly to any section without having to scroll through irrelevant parts of the document. A conventional solution might be to put a list of within-page ("anchor") links at the top of the document so the user can click on the link for the desired section and be taken immediately to it. This can get the user to any section of interest, but requires the user to first return to the top of the page (through a link, by scrolling, or by keystrokes) to go to another section of interest. Putting the list of section links in a frame at the top makes them always visible and available but, as with the content selection control, wastes valuable screen real estate that might better be used to show more of the lesson plan itself. Once again, a control that hides itself when not being used is what best supports the task cases.

Active Table-of-Contents Control

The key step in the derivation of a truly effective solution is the realization that the ideal content selection control and the ideal content navigation control have the same basic structure and closely related behavior. Both must consist of a list of sections that can be selected in some way and both need to hide when not needed yet reappear whenever needed. In the interests of both simplicity and efficient use of screen real estate, a single control that elegantly solves both problems would be desirable. We called the resulting control an Active TOC (Table of Contents) The control is illustrated in the design mockup shown in Figure 1, which is taken from actual working documents used on the original project to document our design. The details of the visual and interaction design of this control are explained in the following sections.

Visual design

For teachers to make effective use of the Active TOC, they need to immediately grasp the dual-purpose nature of the list of contents. We used visual affordance to help communicate both functions to users when they first encounter the control. Check boxes are familiar user interface objects that communicate their own affordance, and nearly all users instantly recognize that they serve to select/deselect individual items to include; this function is also reinforced by the heading at the top of the set of items. (The beveled frame in the illustration of Figure 1 was a later enhancement to improve the visual connection.) To convey the notion that clicking on a label would take the user directly to the section of interest, labels were made to resemble the default form of conventional Web links: underlined blue text. This link-like behavior was reinforced by having the pointer change to a standard "pointing hand" on mouse-over and by a pop-up tool-tip, such as, "Go to Homework section."

Rather than having the check boxes for required contents appear disabled (grayed out), leaving users in doubt as to why these could not be changed, we settled on a unique flag (a colored square) that serves both to distinguish required elements and to visually link to an explanatory footnote, which by its color and positioning draws the eye. The same colored square also appears with each required section itself within the body of the lesson plan, helping to build a consistent mental association and marking required sections as such where they appear within the lesson plan. The order and arrangement of the list of contents in the control also mimics that of the lesson plan proper, further facilitating learning and use.

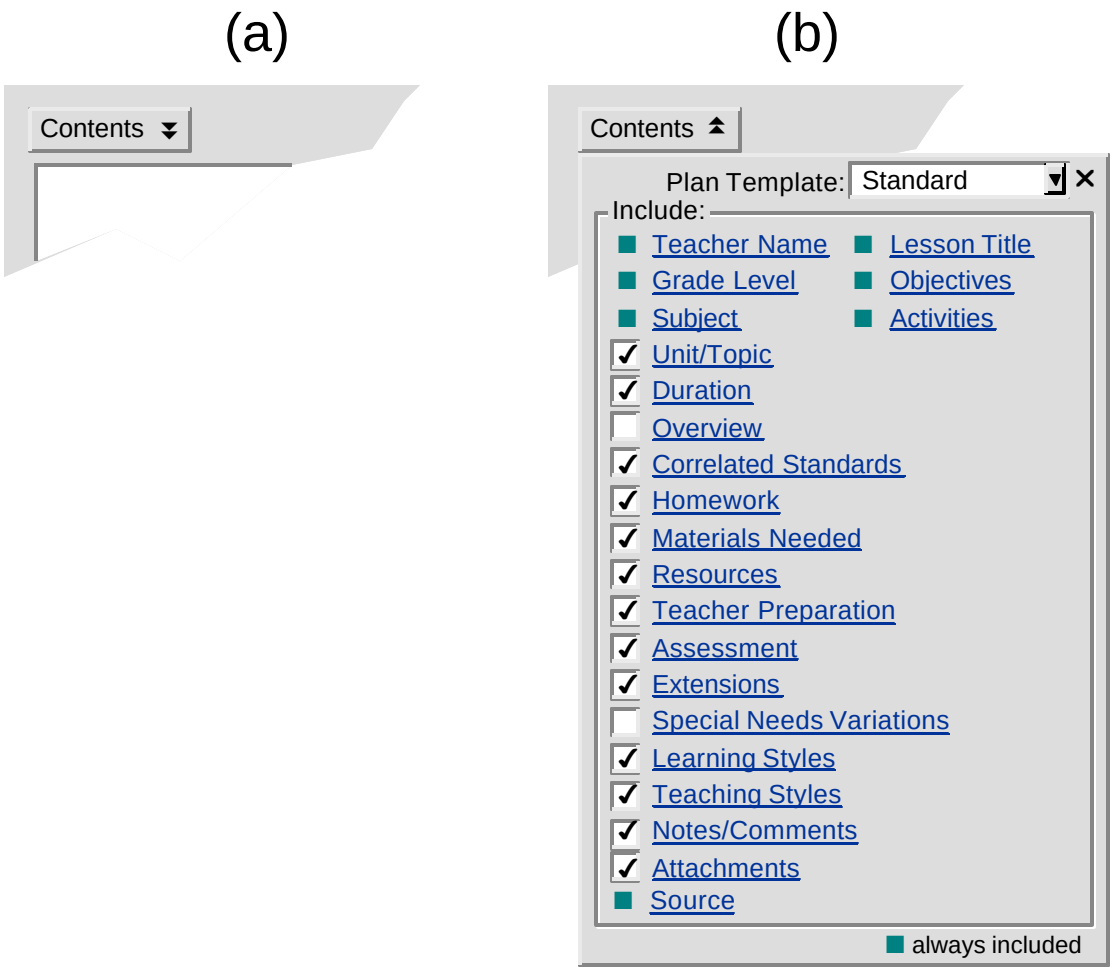

Figure 1 - Mockup for Active Table of Contents control in (a) closed and (b) open conditions.

A conventional drop-down box in the upper right of the Active TOC serves to identify the overall template on which the current lesson plan is based and to select a template when creating a new lesson plan. These templates, defined through a separate process, control the content and format of the lesson plan document itself and the list of contents that appear within the Active TOC.

Interaction design

The control in its closed or inactive state resembles a command button with a double triangle glyph to suggest that something will open downward if it is activated. (We make it appear similar to but different from a standard drop-down because we want users to generalize from familiar components but not expect identical behavior.) Because the Active TOC is so important for efficient user customization and navigation, it is vital that the user discover, understand, and make use of this important capability. However, even with the visual affordance, there is some chance that the new or overly cautious user might not discover this capability through casual exploration. For this reason, the initial state of the control is already open, so that on

first creating or opening a lesson plan, the user would find its table of contents prominently displayed by the Active TOC.

Since the user wants immediate access to any section at any time, the Active TOC needs to be instantly available at all times, which favors a control that, like the Microsoft Office Shortcut Bar, reveals itself on mouse-over rather than requiring an actual mouse-click. The control should remain open to enable the user to perform multiple actions, such as checking or unchecking one or more checkboxes.

Conversely, however, the user wants the control to hide itself automatically when not needed, thus leaving visible as much of the document as fits the screen. Consequently, clicking on a link (section name) hides the Active TOC and scrolls the display to the selected section. The Active TOC also closes automatically after a brief delay if the mouse pointer is moved beyond its boundaries, thus allowing the user simply to begin work without having to explicitly dismiss the control. It can also be explicitly closed by the user, either by clicking on the "close" control in the upper right corner of the panel or by clicking on the button labeled "Contents." This redundancy ensures that new users who did not move their mouse beyond the control's boundaries could still figure out how to "get it out of the way" to proceed with their work or explorations. The presence of the "close" control in the upper right corner also serves to reinforce to the user that multiple actions can be taken within it.

Optimal user-system interaction requires close attention to details, especially of behavior. In this design, resetting a check box to exclude a section does not actually delete it, but only hides it from appearing within the lesson plan body so it can be made to reappear as last defined should its check box be set again at a later time. Users can discover this behavior by trying it, but they get a clue from the tool-tips that appear for check boxes on the control, for example, "Show/Hide Teacher Preparation section."

The proper design of the appearance and behavior for links associated with sections that have been deselected is also important for efficient and natural operation. A programmer-style solution would be to disable (gray-out) such links or, still worse, to present the user who clicks on any link for a deselected section with a modal message box declaring that the section is unavailable because it has not been included.

From a user's perspective, clicking to go to a section that is currently not included has a logical meaning as a natural shorthand for "show/include this section and take me to it." To make it absolutely evident to the user that this is the interpretation taken by the system requires that the user action first result in the check box becoming set, after which the Active TOC closes, and the document is scrolled to the newly inserted or restored section.

Remarks

For user control over contents and rapid navigation within a multi-part document, the Active TOC proved to be an effective solution. It is a remarkably simple and efficient scheme for repetitive use in both capacities, and most users immediately guess correctly at the meaning and function of the control and begin to use it correctly.

Although it is a novel and non-standard user interface "widget," through a design based on instructive interaction, users have little or no trouble understanding and using it. Instructive interaction is accomplished in this case by several small details of

the visual and interaction design, including presenting the control at first without requiring any action on the part of the user. Familiar interface design elements presented with small and deliberate variations in appearance, context, and placement, along with the careful use of visual affordance to convey active functions all help the user to guess correctly at the function and behavior of the various elements of the control.

Beyond its use in the classroom information management system, the Active TOC is a potential solution to a broader class of similar problems in which simple, efficient access to any part of a multipart document is needed along with simple, efficient user control over document contents. Indeed, it is quite likely that other designers bent on similar objectives and solving similar problems have come up with closely related widgets. Even for this particular application, other variations in the visual design are certainly possible and might be as good as or better than the one we devised. Slightly different behavior might also be needed for a good fit in other applications.

The design as presented here is certainly not perfect, and at least one aspect of its non-standard design does create problems for some users. Because the text beside each checkbox is functionally overloaded, serving as both label and navigation link, the user can only check or uncheck by clicking directly on the box itself. If the user fails to understand the visual affordance implied by the link-like appearance of the labels, the result of trying to check or uncheck one of the boxes by clicking on its label can seem very disconcerting and confusing at first. In retrospect, a stronger affordance might be achieved by giving the labels button-like appearance, but only actual testing could ascertain whether this would prove superior in practice.

References

Constantine, L. L., & Lockwood, L. A. D. (1999) ***Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design.*** Reading, MA: Addison-Wesley.

Constantine, L. L. (2000) "Corner Cutting in Model-Driven Web Design," *Software Development, 8* (2), February. Reprinted in L. Constantine (ed.) ***Beyond Chaos: The Expert Edge in Managing Software Development.*** Boston: Addison-Wesley.

Constantine, L. L. (2001) "Management Agility," *Software Development, 9* (6), June.