Customizable Presentations

Tomer Moscovich, Karin Scholz,
John F. Hughes, David H. Salesin

Department of Computer Science
Brown University
Providence, Rhode Island 02912

CS-04-16
October 2004

Customizable Presentations

Tomer Moscovich
Brown University

Karin Scholz
Wolfback Design

John F. Hughes
Brown University

David H. Salesin
Microsoft Research

Abstract
In this paper we explore the possibilities of creating live presentations that can be adapted on the fly in response to the audience and time constraints. Our work also allows presenters to manage multiple versions of a presentation, each customized to a given situation. Customizable presentations are structured by nested sub-paths; this organization has several advantages over linear or hyperlinked presentations. We present a first pass at an interface for authoring and navigating such presentations.

Key words: Presentations, Hypertext, PowerPoint

1 Introduction

Millions of people use PowerPoint and similar tools to create and deliver presentations [7]. Such tools make it easy for speakers to organize their thoughts into linear narratives. But there is a big disadvantage to a fixed linear structure: it disregards the way that talks are typically created, edited, and presented in practice.

Even the best planned presentations often require some clever improvisation as they are given; with current tools, such events are usually dealt with by rapid-fire clicking of the "next" button to skip over sections of the presentation, or by a hand-waving discussion of the topics for which visuals are inaccessible. This disrupts the flow of the talk and consumes speaker time and effort.

Furthermore, it is common to create multiple versions of a talk, intended for different situations or audiences (say, a twenty-minute conference presentation versus a one-hour colloquium versus a five-minute summary). Today, presenters typically create each new version by making a copy of the slide set and editing it for the new situation. As these changed copies multiply and diverge, it becomes difficult to keep all the versions coherent, since edits must be propagated back and forth.

The fact that speakers often append extra slides at the end of their presentations in preparation for questions illustrates that a well designed talk has several paths through it—for example, one for when the audience asks a certain question, and another for when it does not. With that in mind, it becomes clear that version control and dealing with exigencies are two aspects of the same problem. We introduce *structured customizable presentations* as a solution.

In this paper, we present a first interface for authoring and viewing customizable presentations, and for navigating among them while giving a talk. The interface is designed to provide a much greater degree of flexibility to speakers without adding too much cognitive load or authoring overhead to the presenter (who, quite arguably, would rather concentrate on the content of the presentation than on the system for displaying it). Our tools can be used in conjunction with PowerPoint, or with a developing system for creating richly animated slides [10].

A word of caution: Even though a word processor can help you produce an attractive document, it's quite easy – "wizards" notwithstanding – to produce unattractive ones as well; similarly, our interface does not guarantee that a user's exigency plans will be appropriate, or that multiple versions of a talk will form any sort of coherent whole. What it *does* provide is an environment in which well-designed exigency plans and well-related versions of a talk can be maintained and used conveniently.

2 Customizable presentations

Just as a speaker first *designs* and then *presents* a talk, the author of a customizable talk must first design the adaptability and then use that designed-in adaptability during the presentation. In preparing a new talk, or preparing an old talk for a new audience, an author will customize the talk for the expected audience and situation; during the presentation, the speaker may need to adapt the talk to exigencies that arise, such as detailed questions, shortened presentation times, etc. Design and presentation are quite different, and we have two views of a presentation reflecting this.

These two views are designed with several desiderata: first, the simplest customizable presentation should require no more effort to produce than a linear presentation. Second, presenters should be able to add aspects and adaptability to their presentations gradually, each time they prepare to give the talk to a new audience. Third, in the simplest case a customizable presentation should require no more effort to give than a linear one: if things go exactly as expected, the speaker should give the presentation software no instructions other than "advance to the next slide." However, if the unexpected occurs (as

it so often does), the system should allow the speaker to seamlessly shift to a more appropriate version of the presentation or to enhance its contents to suit the audience.

2.1 Structuring a customizable presentation

There are many possible ways to structure a customizable presentation from a collection of slides. The simplest is to give the speaker random access to all the slides. At presentation time, though, the demands of ordering the talk on the fly present a substantial cognitive load, distracting the speaker with the decision of where to go next after each slide. This approach also ignores any logical structure the content may suggest, such as natural sequences of related slides.

Another possibility, one that may be implemented with current technology, is to create hyperlinks among slides, structuring the presentation as a directed graph. This approach lets the speaker build in some sequencing and dependency during authoring, while providing substantial flexibility at presentation time. However, it still requires the speaker to make a choice at each branch point, and as the presentation becomes more complex the demands on the speaker increase.

Ideally, we want a structure that allows the speaker to plan out the order of the talk before it is presented, and also to organize and order alternatives. Our authoring tool accomplishes this through the use of hierarchy and predefined paths. Just as an author might structure a book into chapters and sections, we allow presentations to be structured into nested sections.

Within each section, the speaker defines one or more *section paths*—logical sequences of slides and subsections—represented as paths through a directed graph. A concatenation of nested *section paths* results in a *presentation path*. Unlike hyperlinks, multiple paths intersecting on the same slide each have a predetermined way of leaving that slide, so as long as the speaker chooses to remain on a path, he or she need not make a decision at branch points. However, should the speaker wish to change paths, an intersection of two paths is an ideal place to seamlessly do so.

The key authoring tasks our tool supports are *creating sections* and *authoring paths* through each section. We leave the creation of the slide contents to existing software such as PowerPoint, and focus here on the creation of the presentation structure.

Figure 1 shows the authoring tool, which has been used to impose a complex structure onto a set of PowerPoint slides about subdivision curves, creating a customizable talk. Each box represents a section of the presentation. The system assigns a color to each section, which is later mirrored in the navigation view in order to provide a mnemonic cue associated with that section.

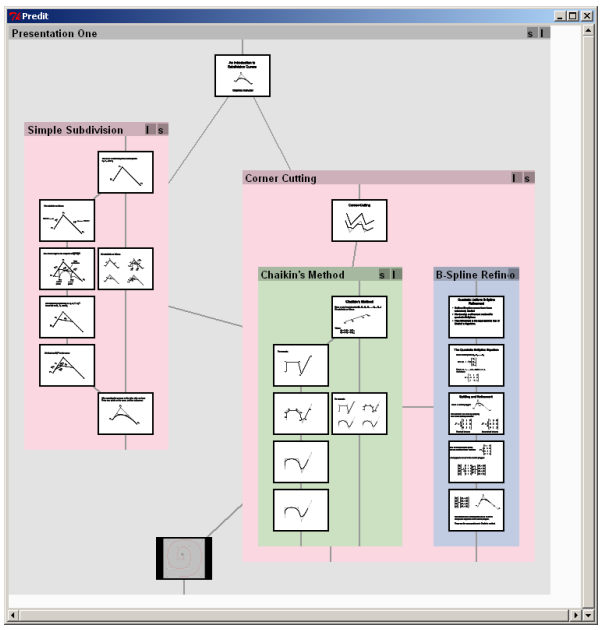


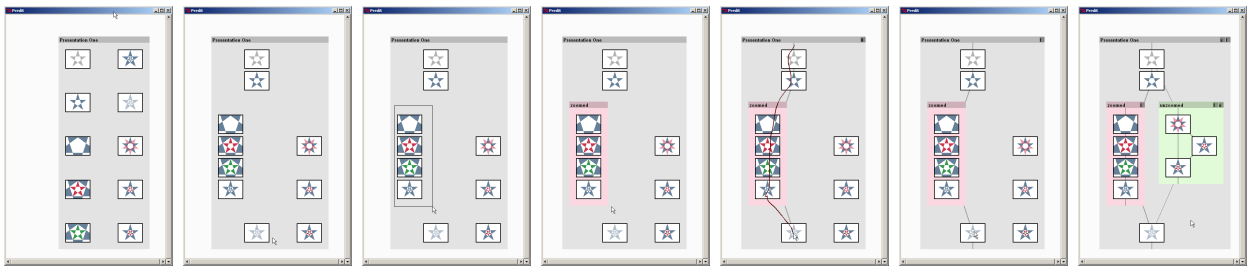
Figure 1: Turning a PowerPoint presentation into a customizable presentation. Here the user has taken a standard PowerPoint deck, and created a nonlinear presentation structure.

Here, the sections serve to organize the talk by topic into two schemes of subdivision, and to segregate the more technical information. Thin gray lines indicate the possible paths through each section. The paths defined in the figure allow the speaker to take either long or short routes through the various sections, as well as to decide whether or not to cover the more technical aspects of the talk.

Figure 2 shows how a user creates a presentation in our authoring system. He or she begins by roughly grouping the slides (each of which is represented by a thumbnail). Then, to define a section, the user draws a box surrounding a group of slides. The surrounded items form a new section. The user creates a path by sketching a curve through a sequence of items in a given section. In our current system, the initial letter of each path's name is displayed in a tab on that section's title-bar. Holding the mouse over one of these tabs highlights the corresponding path.

2.2 The presenter's view

Ideally, navigating a customizable presentation should be as simple as traversing a traditional one; however, the greater complexity of a customizable presentation demands some tool that lets the speaker visualize and control the talk structure. (Even presenters of traditional talks often complain that they have difficulty remembering what slide comes next. Moreover, the large number of



(a) (b) (c) (d) (e) (f) (g)

Figure 2: Arranging a sequence of animated slides into a small multi-path presentation. Each animation is represented by a thumbnail image (a). The user can rearrange the slides (b). By dragging a box around a group of slides (c), slides can be grouped into boxes (d). The user sketches with the mouse (e) to specify a path (f). The completed presentation is shown in (g).

possible choices and actions require more interface than a "next" and "previous" button.) To address these problems, we provide a navigation view: a "big-picture" view of the presentation structure, which is rendered on a separate display from the presentation itself. Dual-headed graphics cards are widely available, even on notebook computers, so this is a feasible approach.

In addition to letting the speaker visualize and control the presentation, a navigation view has several other benefits. Bulleted lists on slides are frequently used for the benefit of the speaker as much as the audience. By showing speaker notes in a separate, private display instead of putting them on the slide, screen area is freed up for illustrations or animations. A navigation view could also provide useful information such as "time remaining" (although ours does not yet do so) and an estimate of how much time a different path through the presentation may take. This would help the speaker make better-informed decisions about improvising during the presentation.

Whereas the authoring view shows a complete structural model of all versions of the presentation, the navigation view focuses attention on a currently active version. Its functions are to help the speaker plan a specific version of the presentation, to track his or her place while delivering the presentation, and to let the speaker improvise changes during the presentation.

The presentation-time visualizer, shown in Figure 3, arranges the active sequence of slides in relation to other possible sequences. The active sequence is highlighted and aligned from top to bottom in a central column, called the *active column*. The speaker can scan down this column visually to preview the slide show. Inactive alternative sequences are shown, dimmed, to the sides of the active column.

To suggest the topology of the presentation with minimal visual complexity, slides and sections are linked by flow lines that represent the union of the possible paths. Thus, paths that share some sequence of slides are depicted by a single flow line. To distinguish and select among available paths within a section, we again use path tabs at the top of each box.

Path tabs corresponding to each path through the section are shown in the right corner of the section's title bar. When the speaker mouses over a path tab, the corresponding path is highlighted and the rest of the presentation is dimmed (Figure 3(b)). Clicking on a tab makes that path active and moves it to the active column (Figure 3(c)). Speakers may use these tabs to rapidly explore presentation possibilities and to control the presentation by changing paths.

Our visualizer can drive either PowerPoint or a script-based slide animation system [10]. When giving a PowerPoint talk, the speaker simply loads the presentation in the visualizer, which in turn launches PowerPoint. Our visualizer controls the PowerPoint application, telling it which slide to display via the Windows COM mechanism. The presenter can advance through the talk along the active path in the usual fashion, by pressing the space bar, or can shift to an alternative path with a few clicks of the mouse.

3 Related work

The canonical software package for producing presentations is PowerPoint [1], which began as a tool for creating offline presentation documents. As such, its presentations are organized in a static linear fashion.

While tools for creating and navigating non-linear documents have been around for decades (indeed, web-browsers have become ubiquitous), few have focused on the specific demands of giving presentations.

The document presentation system of Feiner *et al.* [2] provided for the creation of hierarchical hyperlinked documents. The system was used for maintenance and repair manuals, and could be used to give presentations,

Figure 3: The navigation view for the presentation of Figure 2. (a) The active path is shown as one column down the center. Pressing the space bar or down-arrow will advance along this path. Other paths through the section are indicated by tabs at the upper-right of each section; paths that intersect the current slide have their tab highlighted in pink. Mousing over the tab causes the corresponding path to be highlighted in white (b). By clicking a tab, the user selects a new path; the display rearranges to show the new active path down the center column (c). The space bar will now advance along the new path (d).

but without customizable paths the speaker was forced to invent the presentation sequence on the fly.

Pad [8], an early zooming interface, has been used to give spatially hierarchical talks by navigating to slides in an arbitrary order. This inspired the creation of CounterPoint [3] which supports spatial paths through PowerPoint slides embedded in a zoomable space. It also allows speakers to stray off a path by navigating to an overview, and then zooming in on a new slide.

Zellweger [9] has explored the use of directed paths through hypermedia to reduce disorientation and cognitive overhead. However, this work, and systems based on similar ideas [4], have mainly focused on authoring paths for self-guided presentations where the viewer is not familiar with the material. Thus they do not directly support presentation-time improvisation, nor the mix-and-match customization ability of nested sub-paths.

The Pebbles project [5] studies the use of a PDA as a portable private display for controlling presentations. It allows the speaker to easily jump to a random slide without disrupting the presentation flow.

In a similar vein, the Palette system [6] provides random access to slides from multiple presentations by means of physical paper cards. The system allows for fluid improvisation and furnishes the presenter with a private space in which to manipulate the presentation. While the Palette implicitly supports structures such as slide sequences, it defers all narrative decisions to the time of the presentation. It also imposes an additional burden on the presenter of keeping a physical slide deck in synchrony with the digital one.

4 Discussion and future work

There are many possible ways of representing and navigating customizable presentations. Our implementation occupies a particular point in design space. We allow the presenter more flexibility and control at runtime by representing all versions of a talk as multiple paths though a single pool. We have created several short customizable presentations, but more research is clearly needed into how speakers use adaptability in real-life presentation situations. We offer our tools as test cases against which later methods can be compared.

There are many ways to use today's computational power to help make better presentations. For example, a system could be devised to aid the speaker in choosing paths that fit the time alloted a presentation, or even automatically adapt to the time remaining to avoid going over.

The speaker's navigation view allows for many ways of improving the presentation experience. For example, each time a presentation is given, the system could compile an estimate of how long the speaker spends on each slide. This would allow the system to provide a good prediction of how long each alternative path may take. The speaker may then choose to limit the displayed paths to those that fit the allotted time. Streamlining the navigation view, and allowing the speaker to collapse irrelevant sections would leave more space for extensive notes, a large slide preview, or animation controls for interactive slides. A speaker view that fits on the screen of a PDA would offer complete presentation control from anywhere in the room.

In short, while presentation-authoring tools are well-developed and enjoy widespread use, computer assistance *during* presentations is a wide-open area for research.

References

[1] Microsoft Corporation. PowerPoint 2000. Computer software.

[2] S. Feiner, S. Nagy, and A. van Dam. An integrated system for creating and presenting complex computer-based documents. In Henry Fuchs, editor, *Computer Graphics (Proceedings of SIGGRAPH 81)*, volume 15, pages 181–189, August 1981.

[3] Lance Good and Benjamin Bederson. CounterPoint: Creating jazzy interactive presentations. HCIL Tech Report #2003-03, University of Maryland, College Park, MD 207427, 2001.

[4] Frank M. Shipman III, Richard Furuta, Donald Brenner, Chung-Chi Chung, and Hao wei Hsieh. Using paths in the classroom: Experiences and adaptations. In *UK Conference on Hypertext*, pages 267–270, 1998.

[5] Brad A. Myers. Using hand-held devices and PCs together. *Communications of the ACM*, 44(11):34–41, November 2001.

[6] Les Nelson, Satoshi Ichimura, Elin Ronby Pedersen, and Lia Adams. Palette: A paper interface for giving presentations. In *CHI*, pages 354–361, 1999.

[7] Ian Parker. Absolute PowerPoint: Can a software package edit our thoughts? *The New Yorker*, 2001.

[8] Ken Perlin and David Fox. Pad: An alternative approach to the computer interface. In *Proceedings of SIGGRAPH 93*, 1993.

[9] Polle T. Zellweger. Scripted documents: A hypermedia path mechanism. In *Proceedings of ACM Hypertext'89*, pages 1–14, 1989.

[10] Douglas E. Zongker and David H. Salesin. On creating animated presentations. In *Proceedings of the 2003 Eurographics Symposium on Computer Animation*, pages 298–308, 2003.