NextSlidePlease: Authoring and Delivering Agile Multimedia Presentations

RYAN SPICER, Arizona State University
YU-RU LIN, Institute for Quantitative Social Science, Harvard University and College of Computer and Information Science, Northeastern University
AISLING KELLIHER and HARI SUNDARAM, Arizona State University

Presentation support tools, such as Microsoft PowerPoint, pose challenges both in terms of creating linear presentations from complex data and fluidly navigating such linear structures when presenting to diverse audiences. NextSlidePlease is a slideware application that addresses these challenges using a directed graph structure approach for authoring and delivering multimedia presentations. The application combines novel approaches for searching and analyzing presentation datasets, composing meaningfully structured presentations and efficiently delivering material under a variety of time constraints. We introduce and evaluate a presentation analysis algorithm intended to simplify the process of authoring dynamic presentations, and a time management and path selection algorithm that assists users in prioritizing content during the presentation process. Results from two comparative user studies indicate that the directed graph approach promotes the creation of hyperlinks, the consideration of connections between content items and a richer understanding of the time management consequences of including and selecting presentation material.

Categories and Subject Descriptors: H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia—*Navigation*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Graphical user interfaces(GUI)*

General Terms: Design, Experimentation

Additional Key Words and Phrases: Presentations, authoring, slide-ware

1. INTRODUCTION

Multimedia slide-ware presentations are ubiquitous in business and educational settings where they are used to pitch ideas, convince audiences, and convey educational material. Software such as Microsoft PowerPoint, Apple Keynote, and a variety of similar 'slide-ware' packages allow authors to create virtual slide decks containing text, graphics and audiovisual elements. While slide-ware presentations may run unaccompanied in a kiosk or be distributed in print form, they are primarily intended to serve as a visual aid supporting an oral presentation. The pervasiveness of this approach is underscored by a 2004 Microsoft estimate that 1.25 million PowerPoint presentations are delivered every hour [Mahir 2004].

This work has been supported by the National Science Foundation through IGERT No. 0504647
Author's address: R. Spicer, A. Kelliher, Y. Lin, H. Sundaram, School of Arts, Media and Engineering, Arizona State University, 699 S. Mill Ave., Tempe, AZ. Emails: {ryan.spicer,aisling.kelliher,yu-ru.lin,hari.sundaram}@asu.edu.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

Despite the everyday prevalence of slide-ware supported presentations, scholars and critics continue to raise objections both to the software tools used to author and deliver presentations, and to the content and form of those presentations [Tufte 2006; Farkas 2006; Mignot 2005; Hammes 2009; Gross and Harmon 2009]. Clear opportunities exist to substantially improve the digital presentation experience for presenter and audience alike.

In this article, we present NextSlidePlease, a novel software application for authoring and delivering slide-ware presentations. This presentation support tool introduces several innovative contributions in slide-ware tool design to address challenges identified by users creating and presenting slide decks in support of oral presentations. These contributions address issues of content integration, presentation structuring, time management and flexible presentation delivery.

In this article, we present these characteristics and challenges, and our contributions through the iterative development of NextSlidePlease, a presentation support tool.

1.1 Characteristics

We have previously identified several key characteristics of the authoring and presenting experience [Spicer and Kelliher 2009] and we summarize our findings here. These characteristics are typical of the content, presentation format and author/presenter role. Slide decks are created to support oral presentations. In creating slide decks, authors typically synthesize content from a variety of sources into a single document. In many cases, slide decks re-use content from previously authored decks to create new documents. Reusing material gathered from prior presentations requires careful consideration of coherency in a new or modified presentation context. Slide decks are characterized by a "relentlessly linear" structure [Tufte 2006]; authoring tools have not moved beyond the literal slide projector metaphor.

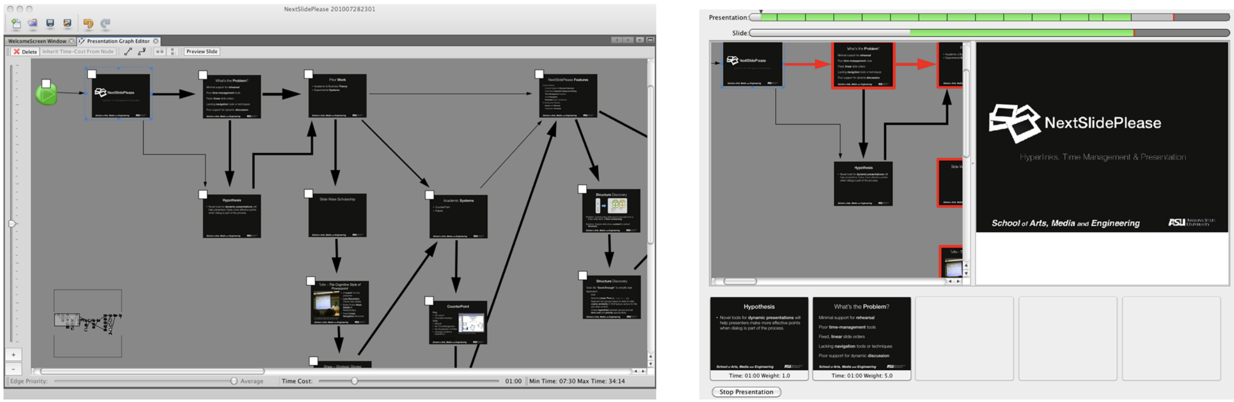
Most slide presentations have known time constraints; many presenters and presentation venues fail to respect them. Slide presentations, particularly when the audience and presenter are at relatively similar levels of seniority within an organization, include question-and-answer periods that break the presentation flow and further complicate time management. Locating relevant supporting material may require the presenter to navigate slide-by-slide through a lengthy linear presentation. These characteristics raise challenges for presenters attempting to convey complex material effectively.

1.2 Challenges

These characteristics of contemporary slide-ware present challenges for both presenters and the designers of future presentation support tools. Some of these challenges include crafting effective narratives, facilitating rehearsal of complex interconnected content, and dynamically balancing audience interests with delivery of required material.

—*Crafting Compelling Narratives that Capture Complex Relationships*: While the linear paradigm used by most widely-adopted slide-ware tools is effective in many cases, such tools force the author to convert potentially-complex relationships into a linear sequence early in the authoring process. This flattening risks losing important connections within the information, and has been cited by critics as a contributing factor to disastrous outcomes in crisis situations[Tufte 2006]. The linear structure also presents challenges for the user in accessing information in response to audience questions. Future slide-ware tools must address the non-linearity of complex information in the informational structure they create.

—*Facilitating the Re-Use of Existing Content and Presentations*: Presenters frequently craft presentation documents by combining many sources of information. These sources of information may include previously-authored presentations. Current widely used tools do not provide computational



(a) The NextSlidePlease Authoring Environment. Users can import slides, create paths/hyperlinks and set time costs.

(b) The Presentation Environment, as displayed to the presenter.

Fig. 1. The NextSlidePlease authoring and presentation environments.

assistance in this process. Future tools must address this challenge and assist users in effectively gathering and shaping information.

—*Effectively Managing Limited Time in Dynamic Presentations*: During presentation delivery, presenters must effectively manage their limited time, covering the full breadth of information while respecting the schedules of their audiences and venues. Question-and-answer periods add complexity and unpredictability, which any tool must account for. This challenge is complicated by the additional complexity that could be introduced by navigating a non-linear structure.

1.3 Contributions

NextSlidePlease, the presentation support tool we have developed (Figure 1), contributes potential solutions addressing these challenges through several novel features.

—*Graph-based authoring*: To facilitate purposeful authoring of coherently connected material, we replace the linear slide deck with a two-dimensional directed, weighted graph. The weighted edges encode two values — the estimated time cost of the target slide, and the relative priority of the target slide. This allows the author to craft a subset of most-likely paths through the slide deck that can be navigated according to time, audience interest or content constraints. This contribution addresses the challenge of authoring compelling narratives that capture complex relationships by allowing the user to encode these relationships into the structure of the presentation.

—*Content Reuse*: To support efficient and mindful reuse of prior presentations, we introduce a presentation structure discovery algorithm. This algorithm reduces the time and cognitive load required to construct a successful graph representation of an existing presentation. The algorithm attempts to discover sections and subsections in content based on text similarity, and construct meaningful edges between them using a heuristic. This contribution addresses the challenge of facilitating effective re-use of existing presentations.

—*Path Suggestion*: To reduce cognitive load on the presenter and to support interactive presentation delivery, we present a path suggestion algorithm. The algorithm considers the current slide, the time-costs and priorities assigned to edges by the author and/or the structure discovery algorithm, and generates a path that maximizes the priority of slides recommended without exceeding the total time budget for the presentation. This contribution addresses the additional navigational complexity introduced by the graph-based authoring paradigm.

—*Time Management*: To help presenters actively manage time, we create a user interface which displays the current slide time limits within the overall presentation time context, including the suggested path from the path suggestion algorithm. This contribution addresses the challenge of effectively managing limited time, with respect to the additional navigation and time-management complexity introduced by graph-based authoring.

Some of the initial motivations and design principles for the system are discussed in an earlier publication [Spicer and Kelliher 2009]. That article presents our 40-participant exploratory survey, which investigated how businesspeople, academics and students use slide-ware. The article also presents results from six semi-structured interviews with businesspeople and academics who use slide-ware regularly.

In this article we present a complete implementation and evaluation of the NextSlidePlease application. The design of NextSlidePlease is informed by the information gathered in the exploratory survey and interviews. In the next section we describe related and prior work in the area. In Section 3, we outline our motivation and a set of guiding design principles. The implementation of the NextSlidePlease application is detailed in Section 4. Section 5 presents a series of evaluation studies and in Section 6 we close with concluding remarks.

2. PRIOR WORK

A variety of applications address the identified weaknesses of current slide-ware tools in the presentation and authoring domains. While these tools offer important innovations, we believe that NextSlidePlease builds and expands the presentation space with real-time decision-making and time management support features. In addition, NextSlidePlease addresses the re-use of existing content and affords presentation in situations where discussion or interactive question-and-answer sessions are likely to require improvisation and thorough knowledge of content.

Originally rising to prominence in the 1940s, corporate and military planners adopted 35mm slide projectors to display text and images in support of oral presentations. The physical mechanism of the projector made presentations an inherently linear sequence of slides — a paradigm that has persisted through to contemporary software tools. The development of cheaper mimeographed transparences opened the presentation field to a wider audience, and introduced a more relaxed delivery structure as the presenter could choose at will from a stack of material. Additionally, dry-erase pens could be used to annotate transparencies during presentation.

The original design document for Presenter (now PowerPoint), references the highly-skilled, labor-intensive process of preparing 35mm slides and transparencies as a primary motivation for the tool [Gaskins 1984]. Presenter aimed to automate the process of typesetting slides for printing; the availability of low-cost LCD projectors in the mid-90s pushed PowerPoint as an end in itself into the mainstream. In addition to PowerPoint, other applications such as Apple KeyNote, Google Presentations, Popplet[1] and Prezi[2] directly afford the creation of slide-ware presentations.

Despite the popularity and utility of slide-ware presentations, several key weaknesses receive considerable criticism. Tufte criticizes the "fixed cognitive style" of slide decks which tends to limit authors to stacks of bullet-point sentence fragments and visual clutter which impedes the clear understanding of data [Tufte 2006].

[1]http://popplet.com
[2]http://www.prezi.com

We consider the development of presentations as a two-part process: *authoring* and *delivery*. This distinction, proposed in [Lanir, Booth and Tang 2008] as "separating content, layout and presentation style" provides a useful framework for discussing slide presentations. In the authoring stage, the user prepares a presentation document. Content is gathered from a variety of sources and structured into a single document. In successful presentations, careful reflection in authoring helps shape complex ideas into a single artifact. Critics argue that the structures built into contemporary tools — bullet point lists and charts with complex default settings — can, in the hands of untrained authors, force content into forms that obscure or alter its meaning. These structures can also encourage authors to remove content entirely to fit the defaults more easily [Parker 2001].

NextSlidePlease contributes to presentation authoring by allowing presenters to structure their presentation as a two-dimensional weighted, directed graph. While the two-dimensional layout approach is used in some contemporary tools [Nelson et al. 1999; Good and Bederson 2002; Moscovich et al. 2004], NextSlidePlease introduces directed edges as a compromise between strictly linear paths through the 2D canvas and completely random-access approaches where the user must manually navigate to the next slide. NextSlidePlease also introduces a presentation structure discovery algorithm informed by information retrieval approaches. This algorithm analyzes the text of existing presentations and segments them to create a two-dimensional graph structure, reducing the complexity of migrating from a linear presentation to NextSlidePlease.

In the delivery stage, the user shares the previously-constructed presentation with an audience. The presenter must make his or her way through the slide deck, covering important material and conveying information to the audience. The presenter must know the content and the structure of the deck in order to respond to audience questions or other interruptions while still respecting the core content of the presentation and time limits. Research suggests that multimedia slide decks created by contemporary tools provide clear advantages, such as affording the inclusion of full-color graphics and creating documents that persist beyond the presentation act. On the other hand, low-tech visual aids such as whiteboards afford more dynamic presentation structure and allow presenters more spontaneity to reply to audience needs. These non-computational solutions do not as easily afford the creation of archival documents [Lanir, Booth and Findlater 2008]. Yates et al. note that "the strong sequentiality [of slide decks] also constrains the presenter's ability to respond flexibly to the local audience's interests and issues," including questions or requests to return to previous slides [Yates 2008].

NextSlidePlease contributes to presentation delivery with a presenter-facing GUI for time management, and a presentation path suggestion algorithm that assists presenters in picking the optimal path to the concluding slide, given remaining time and content priority.

2.1 Presentation Support Software

Several recently developed applications explore aspects of the authoring, presentation and follow-up activities. Our approaches for NextSlidePlease build on the strengths and extend upon the open questions introduced by these applications.

Many recent applications address the need to capture complex relationships among content items and craft compelling narratives. These applications employ both new or unusual hardware configurations and novel software interfaces.

Lanir et al's MultiPresenter application leverages spatial reasoning capabilities to relate content through dual-screen projection [Lanir, Booth and Tang 2008]. Though MultiPresenter retains the traditional linear flow of contemporary tools, it allows authors to create and organize content for both projection screens simultaneously. Users use a presenter-facing display to control the presentation. Users can elect to display the same slide on both screens or to juxtapose content from different slides simultaneously. During presentation delivery, the user can follow the authored sequence or use a pair of scrollable slide lists to select content for each screen. The individual slide lists are similar in form and functionality to those implemented in PowerPoint and other tools.

Lanir et al draw a useful distinction between presentation content authoring systems and systems that support the practice of presentation. Their solution addresses the challenge that, while “slideware systems support dynamic multimedia content, [they] do not as easily support dynamic presentations.” Although NextSlidePlease does not adopt the dual-audience-display paradigm, it does address the need to navigate to content dynamically during the presentation. We utilize a directed graph, rather than one or more slide lists, to allow users to visually create multiple paths through their content and navigate this content during presentation delivery. The two-dimensional spatial navigation paradigm, we believe, offers advantages over the slide list both in terms of authoring and presentation.

Palette, described in [Nelson et al. 1999], uses a barcode scanner to allow access to arbitrary presentation slides, each printed on an index card with a barcode. The physical media approach allows the presenter to select slides at will from a physical card catalogue and advance through the presentation by scanning the cards. This provides a promising alternative to the limited navigational affordances of contemporary slide-ware, and offers the opportunity to preview the contents of upcoming slides by physically spreading out the index cards on a table or podium.

The solely physical dimension of the application, however, presents difficulties in efficiently searching for and locating required information, and requires the presenter to hold a comprehensive understanding of all slides without any hints or support from the application. For this reason, our proposed system allows presentation authors to arrange slides in a computational two-dimensional layout, and to define preferred paths between slides as directed edges. We allow the presenter to click on the slide plane to jump to an arbitrary slide, but offer the preferred paths as a visual memory aid.

Good and Bederson [Good and Bederson 2002] propose replacing the card stack or film strip metaphor with a Zoomable User Interface (ZUI) in their CounterPoint application, borrowing insights from the domain of mind-maps or concept maps [Novak and Cañas 2008] and visual storytelling [McCloud 2000]. This ZUI application uses spatial position and size to assist presenters and users in authoring meaningful, memorable paths through content. Presentation authors can create paths through their content including both slides and zoomed-out overviews of all or a section of the presentation; slides on a selected path are displayed more prominently during presentation. Presenters can navigate to slides anywhere on the two-dimensional map during the presentation but cannot easily switch among these predefined paths once a presentation has started. The application does not provide explicit feedback on the presenter’s position within the slide map, instead relying on the presenter’s spatial awareness, thus increasing cognitive load. Transitions during the presentation are all animated camera-moves through the 2d plane. Neither the authoring nor presentation environment in CounterPoint visually display potential paths between slides, instead relegating this information to a secondary list view beside the spatial canvas. Prezi and Poplett also lack direct visual representation of paths between slides.

This approach — spatial arrangement without visible paths — increases the cognitive load beyond that created by applications using a slide list without offering any way to reduce the complexity. The use of directed edges to encode potential paths through a presentation is introduced as the core contribution of [Moscovich et al. 2004]. The authors of that paper introduce the idea of branching structures, but use a less-flexible visual rendering of presentation structure than CounterPoint, Prezi and Poplett. In their system, the primary flow of the presentation remains vertical, with horizontal tangents providing optional content. The use of arbitrary slide location in a two-dimensional plane permits users to more intuitively grasp the potential sequential relationships between content. This approach leverages the strengths of concept maps as an alternative to linear slide decks [Burkhard et al. 2005].

The Fly application addresses graph-based presentation authoring [Lichtschlag et al. 2009]. The Fly system uses directed edges to describe transitions between "views," which may include both slides and views of the zoomed-out canvas. Fly provides a set of tools for authoring presentations from scratch in a 2d canvas with defined paths. The system does not address presentation.

Another application addresses comparison and combination of existing presentations using a graph-based metaphor [Drucker et al. 2006]. The authors position their system for content creation, leaving presentation delivery to third-party tools. The system uses a graph view to depict relationships between versions of similar slides across related presentation files. Rather than permitting the user to craft a single presentation document that is re-usable in different contexts, the system aims to help users understand the relationship between different documents that they have crafted over time. The generated presentations are saved for delivery in existing slide-ware tools. The tool does not address the need for flexible and improvisational presentation delivery.

None of the previously mentioned tools provide computational support to ease the transition from linear to two-dimensional spatial layout. To address this need, we introduce a computational feature in NextSlidePlease that attempts to discover already-existing structures in the text of imported presentations. This subsystem performs text similarity analysis on the imported slides. The similarity data drives an algorithmic process which clusters slides into related sections and creates edges between these clusters. We term this contribution *presentation structure discovery*.

The presentation structure discovery algorithm draws on the term frequency/inverse-document frequency (TF/IDF) similarity metric [Salton and McGill 1983], taking individual slides as documents and the presentation as the corpus. Segmentation of time-based media based on self-similarity has been considered in prior work [Foote and Cooper 2003] which focuses on audio and video files; our approach differs in that we consider the slide as the atomic unit rather than, for example, audio samples or video frames. The algorithm compares favorably to Outline Wizard [Bergman et al. 2010]. Our algorithm focuses on extracting a single level of segments rather than a hierarchical structure.

While applications such as Palette, CounterPoint, Popplet, and Prezi add significant cognitive load in navigating the 2d presentation graph, none provide advanced time management tools to support the increased complexity introduced by allowing users to select multiple paths during the presentation. Our study of slide-ware use suggests that presenters typically rehearse a presentation once or twice at most, which provides sufficient awareness of time required in the case of a linear presentation; the addition of multiple paths requires a new approach to time management for dynamic presentations.

The NextSlidePlease application introduces a novel time management display, which helps presenters visually understand the current time remaining, and the time management consequences of selecting particular slides. The time management display borrows the form of a timeline. The use of timelines to present a summary of the presentation has been addressed in [Mamykina et al. 2001]. NextSlidePlease adds to this body of work by integrating the timeline view with information derived from the user's presentation graph, including paths suggested by an algorithm which uses the time-budget and priority information encoded in the presentation graph to suggest paths through the presentation's hyperlinks that will meet the presentation's time budget. Our path suggestion approach applies linear programming techniques [Khachiyan 1979; Dantzig 1951] to solve a system of equations representing these constraints. While the techniques used to solve the linear optimization problem are well-established, our formulation of the presentation path suggestion problem as a linear optimization problem is novel. The suggested path is updated in real time during the presentation based on the current slide and time remaining. The time budget is based on the context-specific time-cost assigned to the weighted directed edges. The combination of path suggestion and visualization of time management is not included in any of the other spatial-layout-based presentation systems.

3. MOTIVATION AND DESIGN PRINCIPLES

In this section, we introduce the major design principles behind the NextSlidePlease application. In [Spicer and Kelliher 2009] we presented the results of an exploratory study that investigated how frequent presenters in academic, educational and professional contexts authored and delivered presentations using current slide-ware tools. The paper presented the results of the exploratory study and design sketches for the prototype version of NextSlidePlease. Here we expand on those results by specifying design principles based on those results, describing the implementation of NextSlidePlease (Section 4). We also present the results of two comparative user evaluations conducted with the functional NextSlidePlease application and two contemporary tools (Section 5). Based on those findings, we specify a series of design principles used to develop the current version of the NextSlidePlease application. As the results of the exploratory study in [Spicer and Kelliher 2009] are central to the design of NextSlidePlease, we now review the major findings and present these design principles.

The exploratory study consisted of an anonymous online survey and a series of semi-structured interviews with questions in both instruments focused on how slide-ware software is used to author and present visual aids supporting oral presentations. 40 participants responded to the online survey. Of these participants, 10 self-identified as businesspeople, 9 as academics, and 21 as students. The results indicate that the majority of participants produced presentations once a month or once every few months, but presented at least once a month — presentations must be re-used and re-purposed. Most participants stated that rehearsal was "very important" to their presentation, yet only rehearsed once or twice for a given presentation. The majority of participants who permitted audience questions chose to hold those questions until the end of the planned presentation.

Six participants in the survey were invited to participate in a semi-structured interview. These participants were all businesspeople and academics. In the interview, participants noted that question-and-answer and discussion periods were more likely when the presenter was addressing an audience at a similar level of seniority within a community. Presentations to management, or from management to less senior audiences, tended to avoid interaction. Even within presentations that planned for Q&A, few presenters included back-up slides to address anticipated questions. We focus on the first case, where opportunities exist to improve Q&A/discussion and to better support time management.

3.1 Design Principles

The general design principles of our system are derived from insights from our pilot study. The principles suggested here are additionally grounded in results from the slide-ware literature. We propose the following design principles for information presentation tools that support narrative coherence and adaptable presentation. These tools should:

- Assist the author in prioritizing content based on significance and presentation constraints.
- Support purposeful reuse and modification of prior presentations.
- Encourage rehearsal and content fluency.
- Help the presenter choose the optimal path during delivery.
- Reduce cognitive load on the user; ensure that additional features do not impose undue additional cognitive load.

In the following section, we introduce the NextSlidePlease application, which implements these principles.

4. IMPLEMENTATION

In this section, we first present a system overview of the NextSlidePlease application (section 4.1) and then describe the computational approach used in the system (section 4.2).

4.1 System Overview

The NextSlidePlease application consists of two environments to support the authoring (section 4.1.1) and delivery (section 4.1.2) of a presentation document. Users can switch between the two phases as part of an ongoing cycle of presentation authoring, rehearsal, delivery and re-use. We implement the NextSlidePlease application as a cross-platform Java application on top of the NetBeans Platform[3].

4.1.1 *The Authoring Environment.* The NextSlidePlease authoring environment (Figure 2(a)) allows users to integrate content from multiple presentations into a single artifact and craft paths between imported slides. In the current implementation, the authoring process begins in the user's preferred slide-ware application. NextSlidePlease imports slides in PowerPoint format, or from sequentially-numbered images (with no text analysis function). We do not re-implement basic slide editing functionality in our application to focus more attention on the novel features of the authoring environment.

The Authoring Environment allows presentation authors to reflect on the content of slides and create meaningful associations between that content by spatially organizing slides in a two-dimensional weighted directed graph. The edges and edge weights assigned in this reflective activity provide input to the algorithms used to suggest paths during the presentation. This process also helps the author assess how he/she might skillfully handle questions and navigate while presenting.

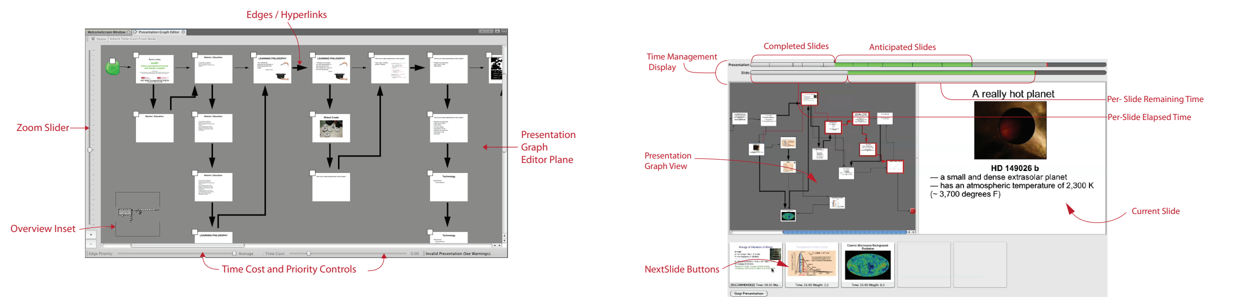
The environment draws inspiration from the previously discussed prior research in zoomable user interfaces for hypertext, concept maps and graph visualization. Our authoring interface displays the presentation as a directed, weighted graph (Presentation Graph). In graph notation, each slide is a node in the graph; each hyperlink between slides is a directed edge. In addition to visually indicating paths, directed edges include two weights: time cost and priority. Time cost refers to the time the presenter believes he or she will require to present the slide at the end of the hyperlink. Priority refers to the importance of following the hyperlink, relative to other hyperlinks from the current slide. Time cost is specified in seconds; priority along a 10-point scale labeled from "Unimportant" to "Mandatory.".

We store time cost on the edge, rather than on the slide itself, to account for context-dependent differences in time requirements. For example, consider the case of a last slide summarizing a section of content. If the presenter selects the entire section for presentation she may spend just a few words on this slide, summarizing the material she already presented. If, on the other hand, the presenter skips the majority of the section, instead displaying only the summary slide (if this particular audience is not likely to be interested in the details, or to save time during a short presentation) she might instead need to spend a minute or two presenting more information. The default behavior allows all edges entering a given slide to inherit time-cost from that slide, thus simplifying editing for cases where the per-edge functionality is not required.

Although the authoring environment provides flexibility for users to establish meaningful structure, connecting slides can be time consuming. We develop an automatic content structure discovery mechanism, described in Section 4.2.1, to overcome this problem.

4.1.2 *The Presentation Environment.* The NextSlidePlease presentation environment provides the user with both high-level 'glanceable' feedback and lower-level navigation control. NextSlidePlease is intended to be used in a dual display configuration, where one display output is connected to, for example, a projector and the second display output (the internal screen on a laptop computer) is visible. The

[3]http://platform.netbeans.org



(a) The NextSlidePlease Authoring Environment. Annotations indicate different components of the user interface.

(b) The NextSlidePlease Presentation Tools View, which displays the current state of the presentation and allows navigation.

Fig. 2. The NextSlidePlease authoring and presentation environments. Annotations indicate components of these interfaces.

Presenter Tools View, shown in Figure 2(b), implements the 'glanceable' and navigation requirements, expanding on interface constructs found in current slide-ware tools. The display consists of four areas: Presentation Graph View, Time Management Display, NextSlide Buttons and Current Slide View.

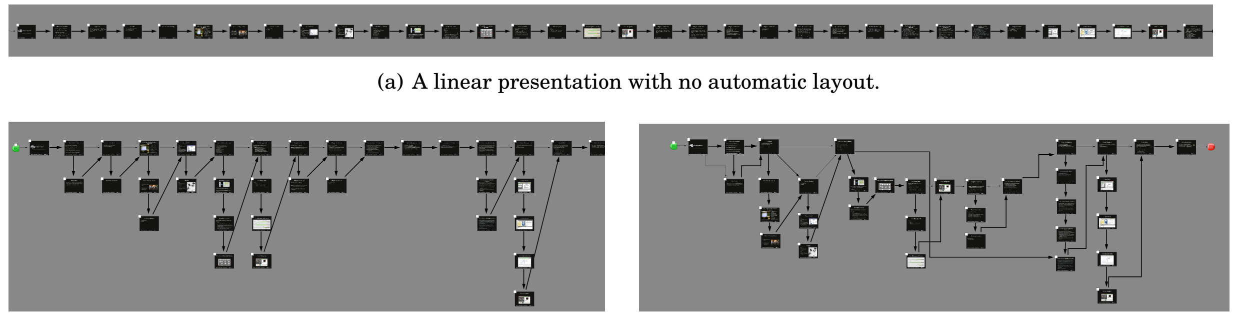
Presentation Graph View: The Presentation Graph View provides the presenter with a visual map of possible navigation strategies through the material. This component displays the Presentation Graph created in the authoring stage. The display highlights the nodes and edges of the currently recommended path. Additionally, the presenter may navigate to any slide by clicking its thumbnail image in the slide map. When the presenter transitions to a new slide using any of these interactions, the Presentation Graph centers on the currently viewed slide and adjusts the zoom level until all slides directly linked from the current slide are visible.

Time Management Display: The design of the Time Management Display is informed by the need to offer the presenter glanceable feedback on time expenditure both at the level of the individual slide and the overall presentation. The Time Management Display, which spans the top of the screen, contains two bar-shaped timeline displays. The Presentation Time-Line displays progress through the entire presentation. Individual slide time budgets are demarcated by black vertical lines. The vertical red bar at the right of the top bar illustrates the total time constraint specified by the user. Space is provided to the right of this bar to indicate over-run, if the presentation cannot be completed in the budgeted time. Already-viewed slides are displayed as white segments; the slides in the projected best path are displayed in green. The Slide Time-Line displays real-time progress through the time budget for the currently selected slide. The green segment represents the remaining time. When 15 seconds remain, the green bar turns yellow to indicate that the allotted time is almost up; after the time budget is exceeded, the Slide Time-Line turns red and begins extending to the right of the time limit line.

NextSlide Buttons: The NextSlide buttons provide a quick way for users to choose between the preferred navigation choices. These choices are offered by a novel path suggestion algorithm, which will be described in the next subsection. Each button displays the thumbnail of a slide linked to the currently viewed slide. The slides are ordered in terms of priority, but the leftmost slide is always the recommended choice. The user may click a button to navigate to the corresponding slide; the buttons are also mapped to the keys 1-5 on the keyboard for one-touch navigation.

Current Slide View: As in other slide-ware tools, this panel displays a scaled-down version of the current slide. This allows the presenter to view that slide without turning to face the projection screen.

During presentation rehearsal and delivery in the Presenter Tools environment, NextSlidePlease generates an XML-formatted log file recording aspects of the delivery. This file encodes which slides



(a) A linear presentation with no automatic layout.

(b) The result of the presentation structure discovery algorithm applied to the same presentation.

(c) The result of the presentation structure discovery algorithm, after editing by a user.

Fig. 3. Results of the NextSlidePlease presentation structure discovery algorithm applied to an actual presentation.

were visited, how long was spent on each slide, and whether the path that the user followed was the path suggested by the algorithm. The logs may be reviewed after the presentation.

4.2 Algorithmic Support

We now describe the computational approach used in discovering presentation structure and suggesting paths through presentation content in NextSlidePlease. These algorithms are described to demonstrate how existing techniques are applied in novel ways to the presentation authoring and path-suggesting domain in the application.

4.2.1 *Presentation Structure Discovery*. The goal of content structure discovery is to reduce the effort needed to connect slide presentations from scratch by extracting relationships among slides automatically. Structure discovery consists of three steps. First, we import a PowerPoint file using Apache Commons POI.[4] This importer allows our application to examine the full text on each slide of the imported presentation. Second, we segment the linear presentation into clusters of slides and create connections between these clusters based on the semantic distance of slide content. Third, we create a basic layout. The user may then modify the presentation graph.

Our segmentation method is based on a simple observation: while typical linear presentation documents do not have explicit structure, the text used in the documents provides clues about how slides are related. Slide-ware presentations are predominantly linear and broken into sections and subsections, therefore the algorithm need only consider the current and immediately subsequent slide. If these slides use similar terms, the slides are likely related in a single section; likewise, if terms are significantly different, the subsequent slide likely starts a new section. To discover this implicit structure, the algorithm uses a feature vector constructed of term frequency / inverse document frequency (tf/idf) metrics [Salton and McGill 1983] for each term that appears in the text.

For the purpose of constructing the tf/idf feature vector, the presentation is treated as the corpus and each slide as a document. A stop list is employed to remove frequently appearing words. We use cosine similarity ($d = (A \cdot B)/(||A|| \cdot ||B||)$) to compare each slide to the slide immediately following it. Two feature vectors are generated for each slide: one using terms in the slide title only, and one using all terms in a slide. For both vectors, fixed thresholds t_{title} and t_{body} are set experimentally. We define the set of slides included in the imported presentation as s. The initial slide s_0 is placed into a new cluster,

[4]http://poi.apache.org

c_0. A segmentation point is inserted between any two consecutive slides, s_i and s_{i+1} if $d(s_i, s_{i+1}) > t$, where t is an empirically determined threshold.

Once segmentation is complete, edges are created from slide s_i to slide s_{i+1} for $i \in [0, |s|)$ where $|s|$ is the number of slides. These edges are assigned a relatively high priority, 7 out of 10, "very important," since they encode the originally intended path through the content. These edges are assigned a default time cost of one minute. Additionally, edges are created from the first slide of each segment to the first slide of the next segment. These edges are assigned low priority, 3 out of 10, "unimportant," since the presenter is unlikely to follow these links except when short on time. These edges are set to a time cost of two minutes, because it is likely that a presenter summarizing content will need more time to add additional description that would have been covered in the remaining slides. This structure of edges formally encodes the strategy "skip a section, except for a summary slide, if running short on time."

The segmentation algorithm works best with presentations that include significant text content. The algorithm does not account for image or video content. Extracting the semantics of images is beyond the current scope of NextSlidePlease. Note that more sophisticated algorithms (e.g. similarity measure based on visual features) may be applied to discover structure in existing presentations. We expose a plug-in API so that additional algorithms may be added and evaluated in future iterations.

The presentation structure discovery process creates a basic presentation graph in the Authoring Environment. This graph presents the detected sections, as vertical columns of slides, arranged horizontally across the 2d graph. This layout visually indicates the extracted relationship among slides. The author may edit this graph to create additional links, destroy unnecessary links, adjust time-cost or priority of any link, re-position slides, and perform additional import operations. Once the user has crafted a presentation graph, the author may rehearse or present it.

4.2.2 *Navigation and Path Suggestion.* NextSlidePlease implements a novel path suggestion and time management algorithm. This algorithm assists presenters in responding to shifting time constraints owing to questions, interruptions and general audience interaction. The algorithm uses a linear programming approach to select a series of slides connected by user-defined hyperlinks that maximize the priority value of all slides visited while adhering to defined time constraints. The algorithm suggests a path based on the presentation context, including the user's time and priority estimate for each transition and the remaining time.

The authoring environment allows users to construct a directed, weighted graph representation of their presentation material, including the estimated time duration and importance for each slide transition. We now describe the path suggestion algorithm in detail. Let $G = (V, E)$ denote that graph where a node $v_i \in V$ denotes the i-th slide, and an edge $e_{i,j} \in E$ from v_i to v_j denote a transition from slide i to j. Let $t_{i,j}$ denote the predicted time duration for slide v_j given preceding slide v_i. We call G a NextSlidePlease graph. Let $b_{i,j}$ denote the priority of a transition from slide i to j. A high value of $b_{i,j}$ means that if i is the current slide, the presenter would prefer slide j to be the next slide.

Let s denote the index of the current slide (source), d denote the ending slide (destination), and C denote the remaining time for the presentation. The algorithm recommends a sequence of transitions from s to d and the total time duration of this sequence needs to be less than C. A recommended sequence may include repeated nodes (except for the end node) but may not include repeated transitions. Our goal is to find a "best tour" that meets the conditions above.

Problem 4.1 (*Best Tour, or BT*). Given a presentation graph $G = (V, E)$ with time $t_{i,j}$ and priority $b_{i,j}$ for each $e_{i,j} \in E$, a current or starting slide index s and an ending slide index d, and a total time duration C, find a best tour T as a sequence $v_0 = v_s, e_0, v_1, ..., e_n, v_n = v_d$, such that for $e_{ij} \in T$, the tour priority $\Sigma_{ij} b_{ij}$ is maximal and tour time cost $\Sigma_{ij} t_{ij}$ is less than C. Note that a tour allows repeating nodes but not repeating edges. The solution to the BT problem gives an optimal tour T where total time cost never exceeds C. However, we note that often the presentation time limit may be exceeded by some margin. Therefore, we consider a relaxation duration δ in addition to the best tour problem.

Problem 4.2 (*Best Tour with Relaxation, or BTR*). Given a presentation graph $G = (V, E)$ as previously described and a relaxation duration δ, find a best tour T as a sequence $v_0 = v_s, e_0, v_1, ..., e_n, v_n = v_d$, such that for $e_{i,j} \in T$, the tour priority $\Sigma_{i,j} b_{i,j}$ is maximal and tour time cost $\Sigma_{i,j} t_{i,j}$ is close to C and less than $C + \delta$. In the following we discuss our solution to the best tour problem. We provide an extended solution to the best tour with relaxation problem in Appendix A.

Best Tour Solution: Our solution relies on formulating the problem into an optimization objective function. The idea is to consider finding a best tour as a special flow problem — we need to traverse edges one by one and cannot traverse more than one edge at the same time. This requirement can be satisfied if (1) only a unit flow is given at the starting node, (2) the unit flow is received at the destination node, and (3) all other nodes on the tour must receive and pass the unit flow, and (4) nodes that pass the unit flow are connected. Let $x_{ij} = 1$ indicate the edge e_{ij} is included in the tour and 0 otherwise. We define an objective with constraints as follows:

$$\text{maximize} \sum_{ij} b_{ij} x_{ij} \tag{1}$$

$$\text{subject to} \sum_i x_{is} = 0, \sum_j x_{sj} = 1, \tag{2}$$

$$\sum_i x_{id} = 1, \sum_j x_{dj} = 0, \tag{3}$$

$$\sum_i x_{ik} - \sum_j x_{kj} = 0, \forall k \neq s, d \tag{4}$$

$$\sum_{ij} t_{ij} x_{ij} \leq C \tag{5}$$

$$x_{ij} \in [0, 1], \forall e_{ij} \in E \tag{6}$$

$$z_{ij} - x_{ij} \geq 0, \forall e_{ij} \in E \tag{7}$$

$$z_{ij} - \alpha x_{ij} \leq 0, \forall e_i j \in E \tag{8}$$

$$\sum_i x_{ik} + \sum_j x_{kj} + \sum_i z_{ik} - \sum_j z_{kj} \leq 0, \forall k \neq d, \tag{9}$$

where t_{ij} and b_{ij} are the context-dependent time duration and priority associated with each edge e_{ij}, C is the total time duration, s and d are indices of the starting and ending slide node, respectively. $\{z_{ij}\}$ are *progression variables* and $\alpha > 0$ is a constant real number which will be discussed shortly. The objective seeks to maximize the total priority of edges. There are several sets of constraints: Equation 2 defines two constraints on the starting node v_s, which means one unit flow is given away from v_s and never returns (ref. Figure 4(a)). Equation 3 defines two constraints on the ending node v_d, which means a unit flow is received at v_d and never goes out (Figure 4(b)). Equation 4 defines a set of constraints for each node $v_k \neq v_s, v_d$, which means nodes other than v_s, v_d are intermediate nodes and their in-coming flow is equivalent to the out-going flow — such constraints are often referred as flow conservation (ref. Figure 4(c)). Equation 5 means the tour's time duration cannot exceed the time limit. Equation 6 requires the solution $\{x_{ij}\}$ to be binary values.

To ensure nodes that pass the unit flow are connected, we introduce a set of progression constraints Intuitively, given a tour T: $v_0 = v_s, e_0, v_1, ..., e_n, v_n = v_d$, we can label each of the edges e_i by some

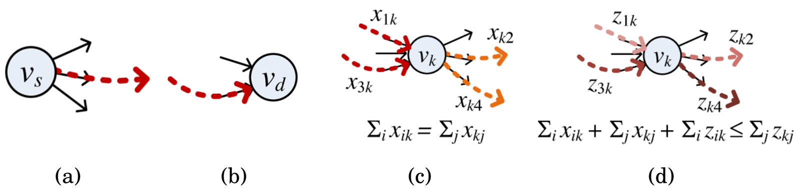


Fig. 4. (a) flow constraints at the starting node; (b) flow constraints at the ending node; (c) conservation constraints at an intermediate node; (d) progression constraints at an intermediate node.

number z_i such that $z_0 \leq z_1 \leq \quad \leq z_n$ show progression from the first to the last edges on the tour. We define such progression variable z_{ij} for each $e_{i,j} \in E$. Each node except for the destination node on the tour should exhibit progression, i.e. the sum of progression associated with its out-going flow should not be smaller than the sum of progression associated with its in-coming flow. We associate the progression variables $\{z_{ij}\}$ with the flow indicators $\{x_{ij}\}$ by Equations 7 and 8; hence for any $x_{i,j} = 0$, and for any $x_{i,j} = 1$, $\alpha x_{i,j} \geq z_{i,j} \geq 0$ — the upper bounds are loose by selecting a large α. Equation 9 defines the progression constraints on each node passing the unit flow. A closed loop without separation from the starting node might satisfy flow conservation but cannot satisfy the progression constraints.

This linear programming problem can be solved in polynomial time [Khachiyan 1979], where the simplex method [Dantzig 1951] and its variants are widely used because they have shown to perform very well in practice. There are $2|E|$ variables and $2|V| + 4|E| + 2$ constraints. In practice, the problem can be solved in $O(M)$, where M is the number of constraints.

The problem might not have a feasible solution. There are two cases: (1) The two nodes v_s and v_d are not connected through any path, hence no tour can be found. (2) There is no such a tour with total time duration less than C. In this case, our algorithm finds the minimal time cost path from v_s to v_d by using a shortest path algorithm (e.g. Bellman-Ford algorithm).

If there exists a feasible solution, we can construct a best tour from the obtained optimal solution $\{x_{ij}\}$ by the following procedure. First, we construct a subgraph $G = (V, E)$ from the original NextSlide-Please graph $G = (V, E)$ such that $\forall e_{ij} \in E, e_{ij} \in E$ iff $x_{i,j} = 1$. Then we find an *Euler tour* starting at v_s from G. An Euler tour is a tour that traverses each edge on the graph exactly once. It can be found by a simple recursive algorithm. Figure 4(a) and (b) illustrate an example presentation graph and its best tour solution for $C = 10$ and $C = 13$.

The additional algorithm to support the Best Tour with Relaxation case expand on the constraints described here and are described in Appendix A.

5. EVALUATION

We have evaluated the system at three points in development. Findings and results from two earlier evaluation studies are summarized in Appendix C. We present here the results of the third most recent user experience evaluation, which directly compared the functionality and end-user experience of NextSlidePlease against two other slideware systems — PowerPoint 2008 and CounterPoint [Good and Bederson 2002]. We also present evaluations of two algorithmic systems included in NextSlidePlease: the path suggestion/time management algorithm, and the presentation structure discovery algorithm.

5.1 Usability Evaluations

This section presents the results of two comparative studies with the current version of NextSlide-Please. The first study compared NextSlidePlease to PowerPoint, a prevalent presentation tool. The second study compared NextSlidePlease to CounterPoint [Good and Bederson 2002], another presentation tool that uses a 2d canvas for spatial layout. The studies asked participants to combine existing slide decks into a single presentation artifact. This constrained task provided a window into how presenters interacted with NextSlidePlease and other slide-ware tools, and allowed comparisons between system usability and end-user experience.

In both studies, participants were asked to provide at least two related slide decks. Participants were asked to use both tools to synthesize the contents of the decks into a single presentation document, which they would then present to the experimenter. All participants were asked to complete both the NextSlidePlease and chosen slide-ware tool conditions; the order of the conditions was randomized to prevent the order from biasing the results in favor of either application. In each case, participants were allowed 30 minutes to author the presentation and between 8-12 minutes to present. Participants were initially asked to target their presentation for a 10-minute time limit. At the start of the first presentation phase, a random number generator was used to select either a 2-minute reduction in time (8 minutes total), the original 10-minute limit, or an expanded 12-minute total. This step was intended to simulate the unpredictable nature of presenting in a business or academic context, where one might be asked to cut a presentation short due to participants arriving late at a meeting or a previous presentation running over. To simulate the case of presenting to ones peers where audience questions would be appropriate, the experimenter interrupted the presenter to ask questions. Since the participants were presenting their own content (potentially unfamiliar to the experimenter), this intervention was somewhat difficult to control across participants.

The order of the conditions was randomized. The same time-limit was used for both conditions for a single participant. This decision was a trade-off between permitting comparison between the strategies used with each tool, and keeping the participant realistically off-balance. This design favors whatever software is evaluated in the second case because participants are already familiar with the task and the content they provided. Randomizing the order of the cases compensates for this. The questions and summary statistics for participants' Likert-scale responses, are presented in Appendix B.1.

5.1.1 *PowerPoint compared to NextSlidePlease.* Six participants completed this study. Each participant provided between two and five slide decks ($\bar{x} = 2.5$, $\sigma = 1.07$). These presentations contained an average of 21.5 slides each ($\sigma = 5.25$). In the NextSlidePlease condition, the presentations created contained an average of 25 slides ($\sigma = 9.9$). These slides contained an average of 33 edges ($\sigma = 15$). This indicates that while most slides were directly connected to only one slide, one out of every 3 slides in the presentation also had a second hyperlink. In the PowerPoint case, the six final presentations created by participants contained an average of 28 slides ($\sigma = 12$). None of the provided slide-ware presentations or the summary presentations created in the PowerPoint case used hyperlinks.

Participants spent the first minutes of the first condition becoming familiar with the material, which in some cases was several years old. The participants then combined slides and allocated time and priority constraints. Participants made comments like "that'll be about 20 slides, but there are a lot of graphical ones..." implying a potential to reduce the time budget estimated. One participant engaged with the automatically generated structure, noting "I can assemble [this set of slides] as an overview along the top [of the presentation graph]," with detail slides below. After several minutes of organization, the participant reflected on some earlier slide deletions: "I didn't really need to delete those," since NextSlidePlease would have allowed him to keep them in reserve. Several other participants, exhibited this delete-first behavior, and later reflected that it would have been wiser to save those slides in reserve. Another participant dealt with the unused slides by deleting all edges, moving them to a corner of the two-dimensional canvas and then re-integrating them as a final step.

Of the six users who presented their NextSlidePlease presentation structures, the users presented an average of $\bar{x} = 19$ slides ($\sigma = 6.6$). This implies that the users skipped approximately 6 slides each.

Of these slides, an average of $\bar{x} = 15$ were presented in the order suggested by the path suggestion algorithm ($\sigma = 6.9$). The presenters finished their presentations $\bar{x} = 50$ seconds faster than their assigned time budget ($\sigma = 130$ seconds). This suggests that presenters are willing to follow the path suggested by the system based on their own composed presentation graph in about 3/4 of cases.

In contrast, none of the participants in the PowerPoint case used PowerPoint's hyperlink functionality. Instead, participants removed slides or, in one case, moved slides they did not require to the end of the presentation order. The Likert-scale responses indicate that the link authoring process encouraged presenters to consider different slide orders they could use for different audiences or situations. $\bar{x} = 4.7$ for NextSlidePlease, compared to $\bar{x} = 2.0$ for PowerPoint (Table I, Question 3).

When presenting in PowerPoint, many participants made comments like "I don't know if I can finish this," referring to completing the presentation in the allotted time. Participants reported that NextSlidePlease's time management function provided more useful feedback than PowerPoint's: $\bar{x} = 4.3$ on a 1-5 Likert scale, as opposed to $\bar{x} = 2.2$ for PowerPoint (Table II, Question 2). Participants also reported that NextSlidePlease's time management function improved their awareness of time remaining: $\bar{x} = 4.6$ versus $\bar{x} = 3.0$ for PowerPoint (Table II, Question 3). This awareness is confirmed by the participants' actual presentation times, as discussed above.

5.1.2 *CounterPoint compared to NextSlidePlease.* In the CounterPoint comparison study, participants were invited to complete the same task: generate a ten-minute presentation summarizing existing presentations and present according to a randomly allocated time constraint. Six participants (different from those in the PowerPoint comparison) completed the study.

The task constructed for the user study favored the hyperlink-centric navigation approach used by NextSlidePlease. CounterPoint requires the user to set a path before beginning a presentation, and does not provide the ability to change paths during the presentation. Due to the duration of the study, it was not possible to ask participants to prepare and deliver more than one presentation in each tool, which might have more favorably highlighted the multiple-paths approach of CounterPoint.

Participants did prefer NextSlidePlease's visual presentation of possible paths to CounterPoint's separation of layout and content order into separate panes. During five out of six participants' CounterPoint presentation, participants expressed sentiments such as "I didn't expect [the presentation] to go there next." One of these users commented "I like that I can see the paths [in the presentation graph view] rather than editing [the CounterPoint path's] slide list to match the [2d] layout."

Table I. . Results from highlighted questions in the authoring section of the two studies' Likert-scale surveys. Note that NextSlidePlease appears to afford increased reflection on the relationships among content in the presentation. No participants in the PowerPoint (PP) case created multiple hyperlink paths, despite the functionality being explained to them. Therefore, answers to questions 4 and 5 in the PP condition are not applicable. Note that otherwise, only responses from participants who responded to questions in both conditions are included in the results.

			PP		NSP			CP		NSP	
Question		n	$\bar{x}$	σ	$\bar{x}$	σ	n	$\bar{x}$	σ	$\bar{x}$	σ
1	It was easy to combine multiple presentations using this tool.	6	3.5	1.4	3.2	0.8	4	2.2	0.5	3.8	0.5
2	Combining the content in these presentations helped me generate new, useful ideas.	6	2.2	1.2	3.7	0.5	4	3.0	1.1	4.0	0.0
3	I considered how long the presentation would take to present while authoring.	6	3.2	1.5	4.0	0.0	4	3.2	0.9	4.2	0.4
4	I considered different slide orders I could use based on the hyperlink structure	6	n/a	n/a	5.0	0.0	4	2.8	1.0	4.2	0.5
5	I could use the authoring environment to create relationships among slides in a way that made sense to me.	6	n/a	n/a	5.0	0.0	4	3.5	1.3	4.2	0.5

Table II. . Likert-scale questions from time management section of the two comparative user studies. PowerPoint (PP) vs. NextSlidePlease (NSP) is presented at left; CounterPoint (CP) vs. NSP is presented at right. Note that scores for NSP are uniformly higher than PP, which includes a simple stopwatch/clock, and CP. CounterPoint does not include an integrated time management tool; in the CounterPoint case, Question 1 was omitted and the other questions refer to a wall clock. As in Table 1, only participants who responded to each question in both conditions are included in the results presented here.

			PP		NSP			CP		NSP	
Question		n	$\bar{x}$	σ	$\bar{x}$	σ	n	$\bar{x}$	σ	$\bar{x}$	σ
1	How often did you refer to the time management features?	5	2.8	1.6	4.8	1.0	4	n/a	n/a	4.3	1.5
2	The time management features provided useful feedback.	5	2.2	1.3	4.8	0.5	4	1.8	0.5	4.3	0.8
3	My awareness of my time budget was enhanced by referring to the time management features.	5	3.0	1.2	4.6	0.9	4	1.8	0.5	4.5	1.0
4	The meaning of the information presented through the time management features was clear.	5	3.2	1.5	3.4	1.7	4	1.8	0.5	4.8	1.2

The Likert-scale data concurs: Presenters reported that NextSlidePlease's integrated approach to displaying both spatial layout and order afforded reflection on different potential orderings: $\bar{x} = 4.7$ for NextSlidePlease and $\bar{x} = 2.6$ for CounterPoint (Table I, Question 4).

Participants did express dissatisfaction with some aspects of dynamic path choice in NextSlidePlease. "I wish I could choose the slides I want to show, and highlight them [while presenting] with a color or symbol," one participant observed. This sentiment was echoed by several other participants. One participant noted "I wish I had fewer choices," referring to the 2d graph view visible during the presentation. Even though she could use several simpler affordances to select likely slides, she said she felt compelled to browse the 2d graph.

CounterPoint does not include any features focused on time management. Participants, when asked to begin their CounterPoint presentation, frequently searched the room for a wall clock. Three presenters checked their cellular phone; two used a watch. One presenter did not consult any timing device, and finished the presentation almost 90 seconds ahead of the ten-minute target. This presenter credits his musical performance training for his sense of timing. Most other presenters finished within about 30 seconds of their target, either rushing through a final slide, or hurrying to finish.

Participants uniformly appreciated NextSlidePlease's time management interface. One participant commented: "The most useful feature is the master time." Other participants shared this opinion, though some suggested other ways to visualize the remaining time. These suggestions involved reducing the update frequency, such as replacing the constantly-updating countdown bar with "time-remaining" cards at various intervals. Another user suggested giving the time remaining in relative terms – "you are one minute behind schedule" on each slide transition.

The positive regard for NextSlidePlease's time management function is reflected in the Likert-scale results (see Table II). Participants reported that NextSlidePlease's time management function provided useful feedback: $\bar{x} = 4.3$ on a 1-5 Likert scale, as opposed to and $\bar{x} = 1.6$ for CounterPoint. Participants reported that NextSlidePlease improved their awareness of time remaining. Note that since CounterPoint does not provide an integrated timer or other time management tool, participants' responses in the CounterPoint case refer to a wall clock, watch, or phone timer.

5.2 Algorithm Evaluation

NextSlidePlease contains two algorithmic systems, as described in sections 4.2.1 and 4.2.2. To validate these algorithms, we conducted studies asking users to evaluate the algorithms in controlled circumstances. Additionally, we report related findings from the user evaluations described above.

Table III. . Results from the path suggestion comparative study. Note that the average scores are uniformly better for the optimal algorithm.

	Question	Optimal		Min-Cost		Max-Priority	
		$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
1	I like to use the algorithm.	3.8	0.9	3.2	1.0	2.7	1.1
2	I think the algorithm is useful in rehearsing a presentation.	3.8	0.8	3.3	1.2	3.1	1.3
3	I think the algorithm is useful in giving a presentation.	3.5	1.1	3.0	1.3	2.7	1.0
4	The recommendations given by the algorithm make sense.	3.5	0.8	3.3	1.2	2.8	1.4
5	The algorithm helps me to achieve better time management.	3.2	1.1	3.7	0.8	3.4	0.9
6	The recommendation given by the algorithm bothers me.	2.3	1.0	2.4	1.0	2.2	1.2
7	I'd rather not use the algorithm.	2.3	1.2	2.5	1.1	2.9	1.4
8	I think the algorithm needs to be improved.	3.1	1.4	3.1	1.1	3.6	1.3

5.2.1 *Path Suggestion / Time Management Algorithm.* To evaluate the effectiveness of our path suggestion algorithm, we designed a tree variant comparison study. In this study, we compared the presentation suggestions offered by our pathfinding algorithm with suggestions provided by two baseline methods. The variants included: (a) Minimize Time-Cost: suggest the lowest time-cost path from the current slide to the end of the presentation, ignoring priority; (b) Maximize Priority: suggest the highest priority path from the current slide to the end of the presentation, ignoring time-cost, and (c) our optimal algorithm, described previously. Eleven participants completed the study.

We asked participants to rate the three variants on eight Likert scale questions. The survey was single-blind, e.g. the participants did not know in which order they were testing the variants, and the variants were not described to them. Some questions were phrased negatively, e.g. "the recommendation given by (*this option*) bothers me." The results from this evaluation are presented in Table III.

Our method outperformed the two baseline options in every question except "the algorithm helps me achieve better time management." In this question, the minimum time-cost algorithm rated highest: $\bar{x} = 3.7$ vs. 3.4 for Max-Priority and 3.2 for Optimal. The optimal algorithm ranked higher than all other options in terms of rehearsal support (Question 2), and presentation support (Question 3).

Although most participants were unfamiliar with the content they used during the experiment, the optimal algorithm received favorable scores. The results suggest our algorithm provides effective presentation support.

5.2.2 *Presentation Structure Discovery Algorithm.* The presentation structure discovery algorithm was evaluated in two ways: a comparative study between human presentation segmentation raters and the presentation algorithm and an end-user evaluation study. Twenty-five sample presentations were collected from five people including a professor, a senior administrator, and three graduate students. Three participants were asked to segment these presentations into sections, where a section is a contiguous set of slides containing related information. We define agreement between users as $a_{i,j} = |s_i \cap s_j|/|s_i \cup s_j|$ where s_n is the set of slides that each rater labeled a segmentation point. To determine average inter-rater agreement between the three raters, we calculate $a_{i,j}$ for each pair of raters and compute the average. To determine agreement between the raters and the algorithm output, we calculate $a_{i,j}$ between each rater and the algorithm, and compute the average. The raters agree with each other with $\bar{x} = 41\%$ and $\sigma = 15\%$; the algorithm agrees with the raters with $\bar{x} = 30\%$ and $\sigma = 12\%$.

An independent two-sample t-test (with p-value=.189) may show that the two measurements are not significantly different. However, due to the small sample size we do not have statistical power to determine the significance. The relatively low inter-rater agreement may be due to the wide variety of presentation content and relatively untrained raters. Overall, we believe that our interview study (Sections 5.1.1 and 5.1.2) provides a complementary, qualitative evaluation about the usefulness of the proposed function. All raters agree with each other on only 17% of slides where at least one rater believes a segmentation point exists. In those cases, our algorithm concurs 75% of the time.

Performance could potentially be improved by adopting more sophisticated machine learning algorithms. However, given the cost of generating human annotated training data and differences in individual authors' styles in balancing image vs. text content, we believe our heuristic works efficiently in suggesting slide segments in an interactive environment for most slide presentations. The algorithm is more effective for presentations with a significant number of words per slide.

This result is supported by comments from participants in the user evaluation studies described earlier in Sections 5.1.1 and 5.1.2. During those studies, participants were asked to rate the algorithmic results along several axes using a Likert scale, and further describe their observations in a semi-structured interview conducted after the usability segment of the study (Reference Appendix B.1). Participants reported that the software accurately captured the dynamics of their presentations in most cases. One participant commented "[seeing the presentation structure suggestion] really draws you in, that it's thinking about how your content goes together." Another participant noted that even though some spurious sections were detected and inserted, the overall experience was good. Several participants throughout the studies noted that they would have liked to apply the algorithm to their presentation after deleting slides made irrelevant by combining/summarizing.

Since neither CounterPoint nor PowerPoint contains a comparable automatic structure discovery tool — the closest analog is CounterPoint's ability to arrange slides spatially based on user-entered outline information — the Likert scale survey questions (see Appendix B.1) cannot be used to directly compare the tools. The scores, however, suggest that our algorithm provides value in triggering reflection on presentation content even if the accuracy of segmentation could be further improved.

6. CONCLUSIONS AND FUTURE WORK

This article introduced NextSlidePlease, a slide-ware application making several novel contributions to the authoring and delivery of presentations. These contributions include: the use of a two-dimensional weighted directed graph to encode the structure of a presentation; a presentation structure discovery algorithm; a real-time path suggestion algorithm; and a timeline management display. These contributions are informed by scholarship in the effectiveness and utility of slide presentations, prior work in slide-ware application development, and our own empirical study of slide-ware usage patterns. Our results across two comparative user evaluations indicate that this paradigm is promising, and suggest provocative directions for future work.

Participants used NextSlidePlease to create hyperlinks between content, generating a much richer description of relationships among content than in their use of either of the other compared tools, where no additional hyperlinks were created. Both studies indicate that users found authoring and navigating with the directed graph-based paradigm more usable than the filmstrip metaphor used by PowerPoint, or the filmstrip and two-dimensional canvas approach employed by CounterPoint.

Promising future directions include: 1) developing and evaluating additional text and non-text based approaches for automated presentation segmentation; 2) developing schemas for the encoding and re-use of effective presentation structures; 3) adapting the presentation graph and time/priority weights based on recorded presentations; 4) developing ways of presenting the captured presentation log to enhance reflection.

The approaches described may also prove relevant beyond the scope of slide-ware visual aids accompanying oral presentations. Similar algorithms and approaches could possibly be employed to author and navigate interactive audiovisual presentations, online instructional materials, or interactive electronic music performances.

REFERENCES

BERGMAN, L., LU, J., KONURU, R., MACNAUGHT, J., AND YEH, D. 2010. Outline wizard: presentation composition and search. In *Proceedings of the 14th international conference on Intelligent user interfaces*. IUI '10. ACM, New York, NY, USA, 209–218.

BURKHARD, R., MEIER, M., SMIS, M., ALLEMANG, J., AND HONISCH, L. 2005. Beyond excel and powerpoint: knowledge maps for the transfer and creation of knowledge in organizations. In *Proceedings Ninth International Conference on Information Visualisation*. IEEE Computer Society, London, England, UK, 76 – 81.

DANTZIG, G. 1951. Maximization of a linear function of variables subject to linear inequalities: in" activity analysis of production and allocation", ed. tc koopmans. cowles commission monograph no. 13.

DRUCKER, S. M., PETSCHNIGG, G., AND AGRAWALA, M. 2006. Comparing and managing multiple versions of slide presentations. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*. UIST '06. ACM, New York, NY, USA, 47–56.

FARKAS, D. 2006. Toward a better understanding of powerpoint deck design. *Information Design Journal 14,* 2, 162–171.

FOOTE, J. AND COOPER, M. 2003. Media segmentation using self-similarity decomposition. In *Storage and Retrieval in Multimedia Databases*. Vol. 5021. SPIE, 167–175.

GASKINS, R. 1984. Sample product proposal: Presentation graphics for overhead projection. Retrieved online http://www.gbuwizards.com/files/gaskins-original-powerpoint-proposal-14-aug-1984.pdf.

GOOD, L. AND BEDERSON, B. 2002. Zoomable user interfaces as a medium for slide show presentations. *Information Visualization 1,* 1, 35–49.

GROSS, A. AND HARMON, J. 2009. The structure of powerpoint presentations: The art of grasping things whole. *Professional Communication, IEEE Transactions on 52,* 2, 121 –137.

HAMMES, T. X. 2009. Dumb dumb bullets. *Armed Forces Journal*.

KHACHIYAN, L. 1979. A polynomial algorithm in linear programming. soviel. *Mathematics Doklady 20, 191 194*.

LANIR, J., BOOTH, K., AND TANG, A. 2008. Multipresenter: A presentation system for (very) large display spaces. In *Proceedings of the 16th International Conference on Multimedia (MULTIMEDIA 2008)*. ACM Press, New York, NY.

LANIR, J., BOOTH, K. S., AND FINDLATER, L. 2008. Observing presenters' use of visual aids to inform the design of classroom presentation software. In *CHI '08: Proceedings of the twenty-sixth annual SIGCHI conference*. ACM, New York, NY, USA, 695–704.

LICHTSCHLAG, L., KARRER, T., AND BORCHERS, J. 2009. Fly: a tool to author planar presentations. In *Proceedings of the 27th international conference on Human factors in computing systems*. CHI '09. ACM, New York, NY, USA, 547–556.

MAHIR, L. 2004. Powerpoint pedagogy. *Business Communication Quarterly 67,* 2, 219–222.

MAMYKINA, L., MYNATT, E., AND TERRY, M. A. 2001. Time aura: interfaces for pacing. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, New York, NY, USA, 144–151.

MCCLOUD, S. 2000. *Reinventing Comics: How Imagination and Technology Are Revolutionizing an Art Form*. Harper Paperbacks, New York, NY.

MIGNOT, J. 2005. Helping engineers and scientists avoid powerpoint phluff. In Proceedings of the 2005 IEEE Aerospace Conference. *, 2005 IEEE*, 4508–4527.

MOSCOVICH, T., SCHOLZ, K., HUGHES, J., AND SALESIN, D. 2004. Customizable presentations. Tech. Rep. CS-04-16, Computer Science Department, Brown University.

NELSON, L., ICHIMURA, S., PEDERSEN, E. R., AND ADAMS, L. 1999. Palette: a paper interface for giving presentations. In *CHI '99: Proceedings of the SIGCHI conference*. CHI '99. ACM, New York, NY, USA, 354–361.

NOVAK, J. AND CAÑAS, A. 2008. The theory underlying concept maps and how to construct and use them. [http://cmap.ihmc.us/publications/researchpapers/theorycmaps/theoryunderlyingconceptmaps.htm], Florida Institute for Human and Machine Cognition, Pensacola, Fl.

PARKER, I. 2001. Absolute powerpoint: Can a software package edit our thoughts. *The New Yorker 77,* 13, 76–87.

SALTON, G. AND MCGILL, M. 1983. *Introduction to modern information retrieval*. McGraw-Hill New York.

SPICER, R., KELLIHER, A., LIN, Y.-R., AND SUNDARAM, H. 2011. User-centered development of a slide-ware authoring tool. Tech. Rep. AME TR 2011-2, Arizona State University.

SPICER, R. P. AND KELLIHER, A. 2009. Nextslideplease: Navigation and time management for hyperpresentations. In *CHI '09: CHI '09 extended abstracts on Human factors in computing systems*. ACM, New York, NY, USA.

TUFTE, E. 2006. *The cognitive style of PowerPoint: pitching out corrupts within*. Graphis Press, New York, NY, USA.

YATES, JOANN & ORLOWSKI, W. 2008. The powerpoint presentation and its corrolaries: How genres shape communicative action in organizations. In *Communicative Practices in Workplaces and the Professions: Cultural Perspectives on the Regulation of Discourse and Organizations*, C. Zachry, Mark & Thralls, Ed. Baywood Publishing, Amityville, NY.

Online Appendix to: NextSlidePlease: Authoring and Delivering Agile Multimedia Presentations

RYAN SPICER, Arizona State University
YU-RU LIN, Institute for Quantitative Social Science, Harvard University and College of Computer and Information Science, Northeastern University
AISLING KELLIHER and HARI SUNDARAM, Arizona State University

A. EXTENDED ALGORITHM

Best Tour with Relaxation Solution: We extend our method to solve the best tour with relaxation (BTR) problem. The idea is that if we can find a best tour without any relaxation, a reasonable BTR solution should not have lower total priority given additional time cost. We formulate the BTR problem based on the obtained optimal solution for BT without relaxation. Let $\{x_{i,j}\}$ be the optimal BT solution for total time cost C, and let $P = \sum_{ij} b_{i,j} x_{i,j}$ be the total priority of the corresponding best tour without relaxation. Similar to x_{ij}, let $y_{i,j} = 1$ indicate the edge $e_{i,j}$ is included in the tour and 0 otherwise. We define the following objective function for BTR:

$$\text{minimize} \sum_{ij} t_{ij} y_{ij} \tag{10}$$

$$\text{subject to} \sum_i y_{is} = 0, \sum_j y_{sj} = 1, \tag{11}$$

$$\sum_i y_{id} = 1, \sum_j y_{dj} = 0, \tag{12}$$

$$\sum_i y_{ik} - \sum_j y_{kj} = 0, \forall k \neq s, d \tag{13}$$

$$C \leq \sum_{ij} t_{ij} x_{ij} \leq C + \delta, \tag{14}$$

$$\sum_{ij} b_{ij} x_{ij} \leq P \tag{15}$$

$$y_{ij} \in [0, 1], \forall e_{ij} \in E \tag{16}$$

$$z_{ij} - y_{ij} \geq 0, \forall e_{ij} \in E \tag{17}$$

$$z_{ij} - \alpha y_{ij} \leq 0, \forall e_i j \in E \tag{18}$$

$$\sum_i y_{ik} + \sum_j y_{kj} + \sum_i z_{ik} - \sum_j z_{kj} \leq 0, \forall k \neq d, \tag{19}$$

where $t_{i,j}$, $b_{i,j}$, C, s and d are defined as above.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

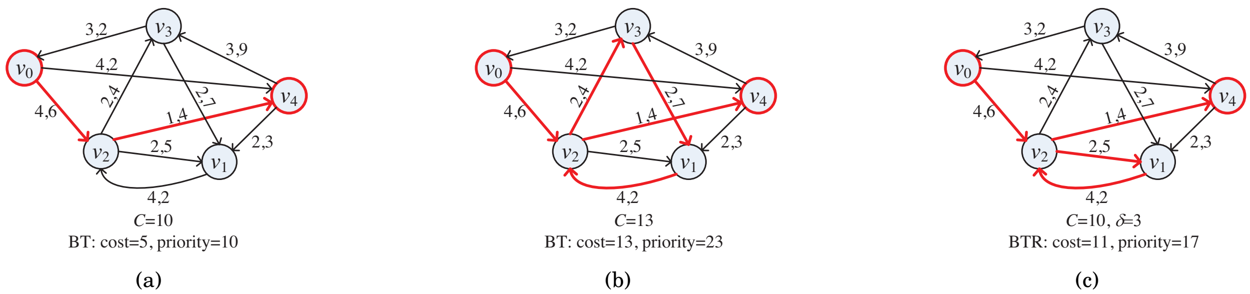


Fig. 5. Example NextSlidePlease graph and the solutions (red lines). Each edge is associated with two variables, time duration (labeled first) and transition priority. (a) Best tour solution for total cost constraint $C = 10$. (b) Best tour solution for $C = 13$. (c) Best tour solution for C=10, relaxation $\delta = 3$.

The objective seeks to find a tour which total time duration close to C, as long as the total priority is no less than P. Hence we are minimizing the total time cost. These constraints are similar to the constraints for BT, except for equation 14 and equation 15. Equation 14 gives both upper and lower bounds for the total time cost of the tour, since the optimal solution with time cost below C is known to be $\{x_{i,j}\}$. Equation 15 means that we only consider solution with total priority higher than P. This optimization objective has the same number of variables as BT problem, with two additional constraints — the lower bounds for both total time cost and total priority of a tour.

The problem might not have feasible solution, i.e. there is no such a tour with total time cost between C and $C + \delta$ and total priority no less than P. In this case the algorithm returns the BT solution since the relaxation does not give a "better" tour. If there exists a feasible solution, we can construct a best tour with relaxation from the obtained optimal solution $\{y_{i,j}\}$ by finding the Euler tour as described above. Figure 5(c) shows a best tour solution for $C = 10$ and $\delta = 3$. Note that the BTR solution is different from BT — when setting $C = 10 + 3$, the BT solution is shown in Figure 5(b).

B. USER STUDY RESULTS

B.1 PowerPoint vs. CounterPoint vs. NextSlidePlease Complete Results

This table summarizes the full results of Likert-scale surveys administered to participants during the PowerPoint vs. NextSlidePlease and CounterPoint vs. NextSlidePlease comparative evaluations. In all questions, the range of responses is 1-5, with 1 corresponding to "Strongly Disagree" and 5 corresponding to "Strongly Agree." Each participant was asked to respond to the same questions for both NextSlidePlease and the other piece of software evaluated. Some questions relate to functions not present in CounterPoint and PowerPoint. These questions were omitted in the CounterPoint/PowerPoint condition, and their absence is marked in the table with a '—.' Please note that no questions were mandatory, and some users did not respond to some questions. In cases where a participant skipped either the NSP or the other-software variant of a question, we have omitted the participant's response to both variants. We acknowledge that it is difficult to draw statistically-valid conclusions from small sample sizes, but we present these summary statistics to highlight differences in participant opinion on the tools.

			PP		NSP			CP		NSP	
Question		n	$\bar{x}$	σ	$\bar{x}$	σ	n	$\bar{x}$	σ	$\bar{x}$	σ
Authoring Environment											
4	I accurately estimated how long each slide would take to present	6	2.7	1.0	2.8	1.2	4	2.5	1.0	3.5	1.3
5	In creating paths between slides, I thought of additional content I could introduce.	6	—	—	3.2	1.3	4	3.2	1.0	3.5	0.6
6	In creating paths between slides, I realized I could remove of make some content less prominent.	6	—	—	4.6	0.9	4	3.0	0.8	3.8	0.5
9	Assigning weights/priorities to each path helped my understanding of the material.	6	—	—	3.2	1.6	4	—	—	3.5	0.6
10	Assigning time limits to each path helped my understanding of the material.	6	—	—	2.8	1.6	4	—	—	3.5	1.3
11	Assigning time limits to each path was useful in my preparation/rehearsal of the presentation.	6	—	—	2.6	1.5	4	—	—	3.8	1.0
12	The meaning of “Time Cost” for an edge is clear to me.	6	—	—	3.8	1.0	4	—	—	4.3	1.0
13	The meaning of each edge’s “priority” is clear to me.	6	—		4.2	0.8	4	—	—	4.0	0.8
Presentation Environment											
18	I was easily able to navigate to the slides I wanted to show next.	6	3.7	1.0	4.0	0.6	4	3.8	0.9	3.5	0.6
19	The software correctly suggested the slide I’d want to go to next.	6	—	—	3.5	0.5	4	—	—	4.3	1.0
20	The suggested path was clear to me from the visual representation.	6	—	—	3.7	1.5	4	—	—	3.5	1.3
21	Observing the selected path gave me ideas about how I might restructure my presentation to meet my time constraints.	6	—	—	3.3	1.2	4	—	—	4.3	0.5
22	Seeing the suggested paths helped me rethink how I prioritized content when authoring.	6	—	—	3.3	1.2	4	—	—	4.5	1.0
23	The consequences of following different hyperlinks, in terms of time management and content covered, were clear to me.	6	—	—	4.0	0.6	4	—	—	3.5	1.2
24	The meaning of the highlighted path was clear to me.	6	—	—	4.0	1.5	4	—	—	3.5	1.3
25	The UI for selecting a slide on a recommended path was clear to me.	6	—	—	4.2	0.4	4	—	—	4.3	0.5
26	The automated layout corresponds to my mental model of the presentation.	7	—	—	3.3	1.5	5	—	—	3.6	1.1
27	Editing the automatically-generated layout helped me think about my content.	7	—	—	4.3	0.5	5	—	—	4.4	0.9
28	The automatically assigned times and edge priorities were useful to me.	7	—	—	3.5	0.8	5	—	—	3.6	0.5
29	The automatic layout tool is useful to me.	7	—	—	3.8	1.2	5	—	—	4.2	0.8
30	It is easy to edit the initially created presentation graph to create the structure I want.	7	—	—	3.7	0.5	5	—	—	4.2	0.4

C. ITERATIVE USER-CENTERED DESIGN USER STUDIES

C.1 Usability Evaluation I

We conducted an initial evaluation of the first-generation NextSlidePlease application. This study was intended to validate the design concepts we had developed and provide iterative feedback for the next generation of the application. The version of the application evaluated in this study included the two-dimensional directed weighted graph and the time management/path suggestion algorithm, but did not include the presentation structure discovery algorithm. The software required that the user manually export slides as images from his or her preferred slide-ware application before use. The Usability Evaluation simulated the process of adapting an existing presentation to NextSlidePlease. Five participants from academia, industry, and K-12 education took part in the study. Participants were asked to provide an existing slide deck in PowerPoint format. Slide priorities were described on a 0–10 scale, instead of the qualitative language adopted later. After creating the presentation, users were asked to deliver the presentation. Observations of participants' interactions with the application during both authoring and delivery were triangulated via observer notes, instrumentation of the software, and user self-report through a 16-question Likert-scale survey and a free-response section. The complete Likert-scale responses are listed in [Spicer et al. 2011].

Results and discussion: The source presentations contained an average of 28 slides ($\sigma = 18$). We categorize the provided documents into three categories: documents presenting internal policy within an enterprise (two presentations), conference presentations (one presentation), and supporting material for academic lectures (five presentations) or informal training (one presentation). The final presentations contained an average of 22 slides ($\sigma = 17$) with an average of 28 edges ($\sigma = 19$). Note that a valid NextSlidePlease presentation graph includes a minimum of $||s|| + 1$ edges, since the slides must be connected to the presentation start and end nodes. This indicates that participants created at least one branching structure in their final NextSlidePlease presentation graph.

In the Likert-scale survey, the majority of users indicated that they found the spatial layout concept promising. Participants confirmed that creating links between slides helped them think about new ways to present their material, and that consciously considering the time cost of different paths was useful in thinking about how to present.

NextSlidePlease would "definitely [be] helpful in a presentation that may facilitate discussion," one participant wrote. The same participant added that "the logic of the paths made me look at the presentation differently." The participants also reported that seeing the path suggested by the pathfinding algorithm was useful in rethinking how they prioritized content in their presentations.

Several trends emerged from the observations. First, users were excited by the ability to visually arrange slides and hyperlinks, and spent significant time investigating how they might re-order their slides. Second, the user interface was mostly considered straightforward. Users did, however, express confusion about the meaning of the numerical priority values. This version of the software expressed path priority on a scale of 0-10; future versions were modified to use subjective text labels like "Unimportant" and "Extremely Important" in response. One specific question, "I had more awareness of time remaining, compared to PowerPoint or KeyNote," averaged a score of 3.8, indicating that the time management features helped presenters remain aware of their time situation.

Several participants noted that while the the process of converting an imported linear presentation into a presentation graph provided useful insight into the way their presentation was structured, creating the graph required significant time and effort. To address this challenge, the presentation structure discovery algorithm described previously was created. This subsystem was evaluated in the second and third user studies.

C.2 Usability Evaluation II

A second usability evaluation was conducted using the next version of the NextSlidePlease software, improved based on feedback from the initial study. Nine participants took part in the study. Of these participants, four self-reported that they had participated in study one. In this iteration of the application several key features had been added: (a) support for importing presentations directly; (b) implementation of a presentation structure discovery algorithm to reduce the complexity of importing presentations; (c) modification of the priority slider control to use descriptive text labels rather than a numbered scale. As in the initial usability evaluation, the summary statistics for the Likert-scale survey are available in [Spicer et al. 2011].

The presentations provided by participants in the study contained an average of 29 slides ($\sigma = 14$). The NextSlidePlease presentations created contained on average 22 slides ($\sigma = 9$) and 29 edges ($\sigma = 11$). This indicates that the structure discovery algorithm encouraged users to create several additional paths per presentation. Potentially confounding factors include increased familiarity with NextSlidePlease and the ideas presented here by the four repeat participants.

Participants who self-identified as participating in both studies uniformly preferred the second iteration. Several participants stated that, despite the relatively naive segmentation algorithm, the automatic layout feature accurately generated a representation of their mental model of the presentation.

Two participants argued that the automated layout function was not useful in its current form. One of these participants identified as a visual artist, and stated that they would rather start from a basic linear slide order and compose linkages from scratch. This user requested that the automatic layout feature be made optional. The second participant argued that because his or her authoring style favored slides with minimal text, text-based similarity metrics were ineffective at segmenting the presentation.

Several new participants made statements like "I can see how this tool could be very useful for organizing presentations, since I build a lot of new presentations based on older slides." This participant is among those who suggested a workflow for path creation.

One user, who self-identified as a music teacher, drew parallels in style between presenting in NextSlidePlease and structured improvisational music styles such as jazz. Our observation of this user's presentation flow supports this parallel. Several other users with less formal training displayed similar behavior. In contrast, two users stated directly that they were relying on detailed knowledge of their material rather than the computational tools for time management. Many users did not manually adjust slide transition priorities in this iteration, even when adjusting the automatic layout. This suggests that the importance of priority is unclear to users. One repeat user noted that the defaults created by the new importer seemed acceptable for the purposes of this study but raised questions about whether or not it would be effective in everyday use.

This evaluation continued to validate the promising nature of NextSlidePlease, but also reaffirmed that a more ecologically-valid study design addressing presentation was required. The longer-form, comparative studies inspired by this need are described in Sections 5.1.1 and 5.1.2.