

Anecdote : A Multimedia Storyboarding System with Seamless Authoring Support

Komei Harada, Eiichiro Tanaka[†], Ryuichi Ogawa[†] and Yoshinori Hara

C&C Research Labs., NEC USA, Inc.

[†]Information Technology Research Labs., NEC Corporation

{harada | hara}@ccrl.sj.nec.com

[†]{tanaka | ogawa}@joke.cl.nec.co.jp

ABSTRACT

In the development of a large-scale multimedia application, the design of the overall scenario structure is as important as the quality of each media content. With current authoring systems, authors cannot start the authoring process and proceed until the media contents are created. Moreover, the overall structure of the application cannot be evaluated until the end. Also, there is not enough support in the early-stage design phase for flexibly constructing the framework and meeting the size/speed limitations of the final application.

We have developed an authoring system called *Anecdote* in order to support the early-design phase as well as the whole development process of multimedia applications. *Anecdote* employs the concept of *surrogate media* and *surrogate scene*. Surrogate media is a type of media, such as sketches, to represent the content of the media data to be created. Surrogate scene can be a design sketch of a screen representing a segment of the application structure. Both can be used to simulate the execution of the scenario and can serve as the specification of the data to be created. *Anecdote* allows authors to choose from different authoring styles (hypermedia/linear, graphical/textual, etc.) to construct the scenario framework, and helps them to create the final application seamlessly from the prototype scenario. Through the development of sample applications by professional designers, we have demonstrated the usefulness of the authoring design principles of *Anecdote*.

KEYWORDS

seamless authoring, surrogate media, surrogate scene, timeline path, rapid prototyping, hypermedia design, sketching interface

Permission to make digital/hard copies of all or part of this material for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copy-right notice, the title of the publication and its date appear, and notice is given that copyright is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires specific permission and/or fee.

ACM Multimedia 96, Boston MA USA

© 1996 ACM 0-89791-871-1/96/11 ..\$3.50

1. INTRODUCTION

Due to the recent progress of multimedia technologies for high-performance platforms and widely accessible distribution media such as CD-ROM's and the Internet, a great number of large-scale multimedia titles are being created. In order to develop a qualitative multimedia title, the well-balanced overall design is as important as the quality of each media content. However, current multimedia technologies fall short of the early-stage design support that ensures the overall design and performance of the application.

When professional designers create the design of an application, they often draw sketches on paper as in **Figure 1**. These sketches are useful for visually describing the screen image and user interaction design. We have developed a multimedia authoring system called *Anecdote* to introduce this flexible user interface of sketching into the early-stage design support. We propose the concept of *surrogate media* and *surrogate scene* as an abstraction of the sketch objects in order to enhance electronic sketching

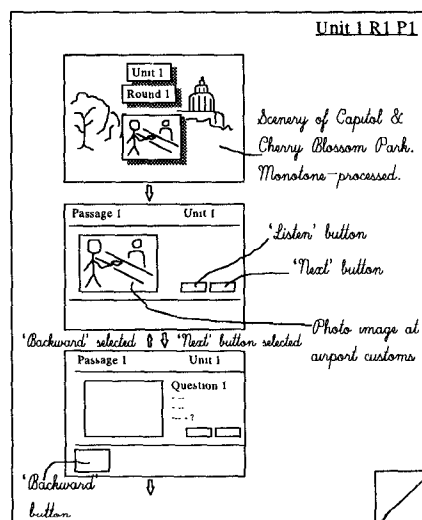


Figure 1: Example of storyboard

interface into integrated authoring design support. With this concept, *Anecdote* provides support for various authoring styles, such as top-down and bottom-up, as well as the support for the seamless development process of multimedia applications.

Related work

Our research started from the need for early-stage design support we observed when we developed the CD-ROM based English Listening Course using our *Videobook* authoring system. The scenario structuring method employed by *Anecdote* is an extension of the design strategies discussed in [10] and [11].

Our work was also influenced by researches in rapid prototyping for visually creating user interface design without programming languages. The techniques explored in this field include: state-transition representation used by Statemaster (Wellner [12]) and PBD (Modugno, *et al.* [8]), the spreadsheet model proposed with NoPumG (Lewis, *et al.* [5]) and the constraint-based approaches studied by Kurlander, *et al.* [2] and MacNeil (TYRO) [6]. As for the design process of multimedia applications, MacWeb (Nanard *et al.* [9]) discusses the support for flexible hypertext design and generating actual node and link structure from the model. However, they do not grapple with the authoring process of multimedia applications in which the media contents require a lot of development time and storage space. They also do not discuss intuitive user interface for multimedia scenario design.

SILK (Landay, *et al.* [3, 4]) considers flexible sketching interface to create prototype and transforming it into the final application. However, they focus on the sketching interface for creating a user interface with widgets (scrollbars, menus, *etc.*) and recognizing such widgets to simulate their behaviors. They do not discuss creating and importing multimedia data and thus changing the prototype smoothly into the final application.

2 DRAWBACKS OF CURRENT AUTHORING SYSTEMS AND METHODOLOGIES

2.1 Problems of current authoring design process

When authors select a current authoring system (*e.g.* ToolBook[1], Director[7], Authorware [13]) as the target platform and start the design of the application, authors must (1) create media contents, (2) import media contents into screen, (3) design the screen layout, display timing and user interactions within the same screen, and then (4) create different screens and define presentation flow and hypermedia links. Therefore, authors usually cannot evaluate the overall design of the application until the development is almost finished. It is difficult to verify the

scenario structure¹ in the early stage of the development. Furthermore, it is often difficult to create media contents first without knowing the context in which they are used in the application, and the design process cannot start smoothly when no media contents exist.

In the development of large-scale applications, authors often create a “mock-up” using a prototyping tool to design the framework of the application. In this case, authors must start the editing process all over again on the target authoring system when the design is complete. And, the application often diverges from the mock-up prototype because the changes made to the former are not directly reflected in the latter.

2.2 Rigid input format and lack of scenario structuring support

When designers create the design of a multimedia title, they often prefer pencil and paper to make design specifications. One of the reasons is that they do not want to be forced to “think” within the input format employed by the authoring system. There are page-based systems (*e.g.* HyperCard, ToolBook), timeline-based systems (*e.g.* Director) and flowchart-based system (*e.g.* Authorware), *etc.* In general, page-based format is appropriate for visually representing each state of execution of the scenario, but not sufficient for media synchronization with video and audio being played throughout more than one pages. Timeline format is not appropriate for visually editing the navigational flow and structure of the application. Flowchart format can specify the media display sequence and user interactions in detail, but each state of execution is not explicitly displayed and cannot be edited intuitively.

2.3 Lack of early-stage feedback to meet size limitations

While most large-scale multimedia titles have been distributed on CD-ROM's, the network browser such as the World Wide Web, is also becoming an important distribution medium. When designing a CD-ROM title, the limit of total data size is a big factor to be taken into account. As for the network, the limitation is on the average and peak data transfer rate. Therefore, the data size of each screen is more important. Although these limitations are a big concern for multimedia title designers, currently available authoring systems do not provide any feedback to meet these limitations in the early-stage design phase.

¹ In this paper, the term *scenario* refers to the framework of the application associated with user interaction and presentation flow, generally as opposed to the media contents.

3 AUTHORIZING DESIGN PRINCIPLES OF ANECDOTE

In order to provide the flexible sketching scenario design interface and seamless development process, we propose the following key concepts.

3.1 Seamless development from prototype to final application

Surrogate media

When authors are designing the application and the media content already exists, authors can simply import it into the application. Otherwise, authors can draw sketches and/or input text annotations to describe the media content. We call such sketches and text annotations *surrogate media*. Surrogate media is used to construct the “mock-up” framework of the application and to enable rapid prototyping. Authors can use surrogate media in place of the actual media contents to simulate the execution of the scenario.

Surrogate media also serves as the specification of the media content to be created. It helps authors to identify each media content used in different parts of the application and to understand the context in which it is used. When the actual media contents have been created, authors can replace each surrogate media with media contents and change the mock-up prototype into the final application smoothly.

Other types of media data can be surrogate media if it can be inputted more easily than the actual media content to be created. For example, authors can record their own voice as surrogate media for audio data. They can hum into the microphone to describe the music, or say “ta-da!” to describe a sound effect. They can use existing clip art files in place of high-resolution photo images to be created. A 3-D wire frame image can pass for the corresponding rendered image, and a series of still images or sketches can represent a video file.

Surrogate Scene

Similarly, we introduce the notion of *surrogate scene*. Surrogate scene is a schematic representation of a certain segment of the application. For instance, one design sketch of a screen explaining the Eiffel Tower can be used to represent the “Paris Tour Guide” section of a World Travel Guide before many screens and detailed navigational flows are created. Authors can similarly create surrogate scenes for other cities, execute the scenario and evaluate the navigational flow from the main menu to each city.

Rapid prototyping using surrogate media and surrogate scenes

Anecdote provides sketching interface to input sketches with text annotations as the main type of surrogate media. One of the design principles of the sketching interface of

Anecdote is to enable authors to input their designs smoothly and to construct the overall scenario framework as quickly as possible. To that end, *Anecdote* allows authors to input the scenario roughly first, and later helps authors to refine the details of the scenario. For example, *Anecdote* keeps track of how many times each (surrogate) media object has been moved or resized on the screen. Authors can design the screen layout coarsely with sketches, and later, they can pick up the scenes that potentially need more polishing of the layout. *Anecdote* also helps authors to find the media objects with attributes not yet inputted.

However, for particular scenes, authors may want to specify detailed layout and display sequence. For instance, in the opening scene of an application, transition effect is important and authors want to input and try different effect types. When *Anecdote* executes the scenario, media display attributes such as effect types are also applied to surrogate media so that authors can realize the behavior of the scenario in the early stage. Such attributes are preserved when the surrogate media are replaced with actual media contents. Authors simply have to replace surrogate media one by one and they can seamlessly develop the prototype into the final application.

3.2 Support for various input formats and scenario structuring

The scenario format as in **Figure 1** is often called *storyboard*. In a storyboard, each frame of sketches represents a state of scenario execution, and the arrows between frames are (often implicitly) associated with an event such as user input, end of media playback, timer elapse, etc. By associating frames and arrows to nodes and links respectively, the storyboard scenario can be naturally represented as a hypermedia structure. The structure of the whole application is organized as a hierarchical hypermedia so that authors can organize the large-scale, complex structure of the application using both top-down and bottom-up approaches

In a scenario represented as a hypermedia structure, there is usually a sequence of nodes and links forming a “main stream” that covers most of the nodes. Other links are considered additional branches coming out of this sequence. In order to allow authors to edit the linear aspect of the scenario, we define this sequence as *timeline path*. Timeline path is automatically defined by the authoring system and can be changed easily by authors. By using timeline path, authors can view the scenario both as hypermedia structure and linear presentation. The user interactions as well as media synchronization can be edited using the hypermedia format, timeline format and graphical/textual outline formats.

3.3 Scenario analysis, estimation and scheduling

When authors construct the framework of the application using surrogate media and surrogate scenes, various characteristics of the final application are estimated based upon the data already inputted. For example, the approximate data size for each scene and the whole application can be calculated using the size attributes for each surrogate media. The sizes of surrogate media are known when authors create them on the screen. Therefore, if authors input additional attributes such as the bit depth for bitmap images and the length and sampling rate for video and audio, the total data size can be easily calculated. The authoring system can detect when the data size exceeds total size limit or transfer rate limit, and advise the authors to reduce data size.

The authoring system can also help authors to estimate the time and cost needed to complete the development. When authors create a surrogate media or surrogate scene, authors can input index attributes of estimated complexity (e.g. "simple (less than 10 min)," "complex (up to 2 hrs)," etc.) and get the rough assessment of time and cost needed to complete the development. These estimated figures are very rough, but still have great importance in the early design stage.

If authors input the start date and end date of the development project, the authoring system can compare the current status and the estimated final status of the application, and analyze the progress of the project. For instance, the progress of media content creation can be known by the number of surrogate media and the number of media contents already created. By comparing the progress of different parts of the application, authors can detect potential bottlenecks of development.

4 IMPLEMENTATION

Anecdote is currently being developed on Microsoft Windows platforms. This section describes *Anecdote*'s scenario structure and the editing views, and a typical authoring process using *Anecdote*.

4.1 Scenario structure

With *Anecdote*, the application is constructed as a hierarchy of *scenes*. An object called *group scene* can be used to modularize the lower levels of scenes and other group scenes. Each scene represents a state of scenario execution and is defined as a set of *slots*, which are the place-holders for the media data, or *cast members*, to be presented.

All types of media data are managed as cast members. A cast member is created by importing a media content or creating a surrogate media. The surrogate media is still retained in each cast member even when they are replaced

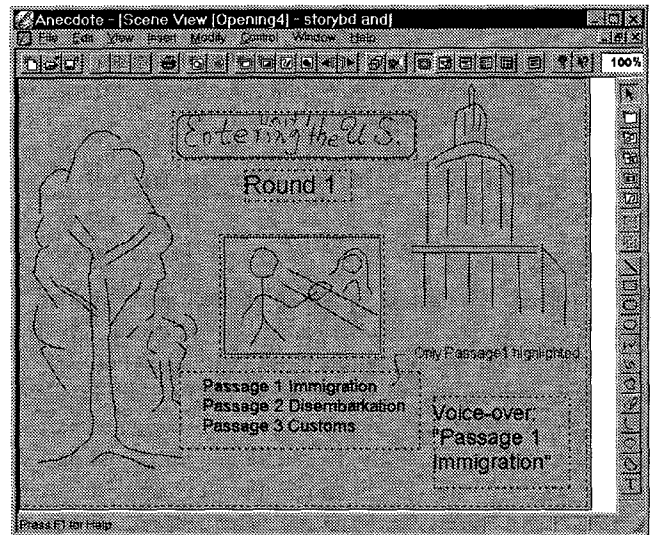


Figure 2: Sketching interface of the Scene View

by the actual media content. Therefore, authors can switch back to "sketch mode" at any time to see the original specifications. A single cast member can be referred to by slots in different scenes. Therefore, copying the scenes to describe different states of scenario execution in detail does not significantly increase the total data size of the application.

A *link* connects one slot in a scene to another scene. It is associated with a *user event* such as mouse and keyboard input, or a *system event*, such as end of media playback, timer elapse, etc. By specifying a user input, any type of visual media can become a "button."

4.2 Editing views

In order to edit different aspects of the scenario, the *Anecdote* Storyboarding system provides five editing views, namely the *Scene View*, *Link View*, *Timeline View*, *Outline View* and *Cast View*.

From within any of the views, authors can choose "Play" menu to execute the scenario being edited. The scenario is executed with surrogate media and/or the media contents already created.

With *Anecdote*, all views can be printed out. For instance, the hardcopy of the Cast View with surrogate media becomes the specification document for the media contents to be created. In a large-scale, collaborative development, hardcopies are essential because the persons who create media contents can be different from the designers, and they can be out in the field when they get the media materials.

Scene View

In the Scene View, authors can edit the screen layout of a scene (Figure 2). Authors can create a slot by importing a

media content, copying a cast member from the Cast View, or creating an empty slot and inputting sketches and text annotations.

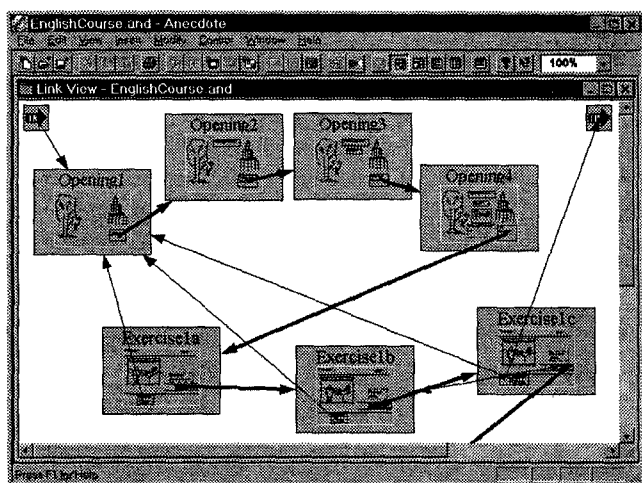


Figure 3: Link View

Link View

In the Link View, each scene is displayed as an iconized image of the screen layout (Figure 3). Authors can create links by dragging the mouse from one slot to another scene. Authors can also edit the hierarchical structure of the scenario using group scenes.

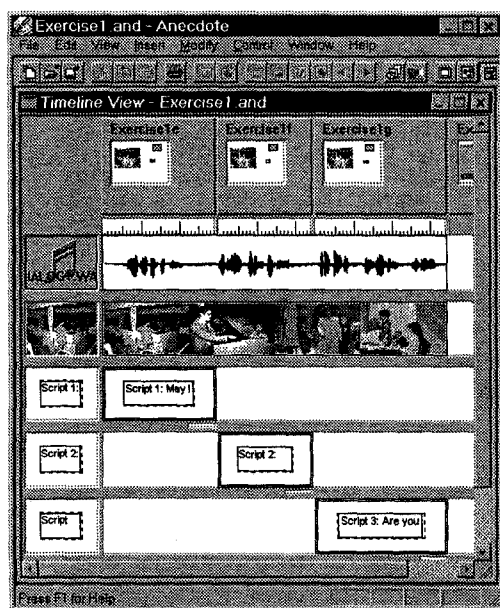


Figure 4: Timeline View

Notice the script texts are synchronized with the audio.

Timeline View

In the Timeline View, scenes are displayed from left to right in the order they are to be executed (Figure 4). Cast members used in those scenes are displayed as a timeline. The frames of video and wave form of audio are displayed inside the timeline.

Outline View

In the Outline View, scenes are displayed in the left from top to bottom (Figure 5). Slots in each scene are displayed next to the scene with their names, text annotations, file name, and attributes. Authors can edit scene sequence as a graphical storyboard or textual outline.

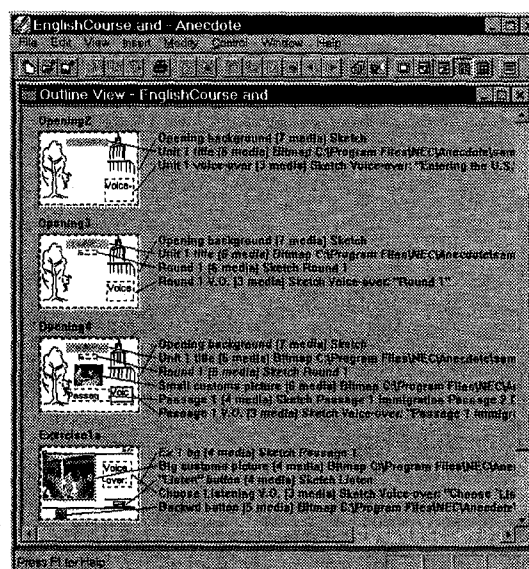


Figure 5: Outline View (Graphical mode)

Cast View

In the Cast View, the cast members used throughout the scenario are displayed. Authors can create new cast members by importing media contents or creating surrogate media. The display can be sorted by the default media type and the user-defined attributes authors have inputted.

4.3 Progress monitor

Progress Monitor shows the following information to the authors:

Analysis: The current status of the scenario, such as number of scenes, group scenes, slots, cast members, and the update history of each scene, slot, cast member, etc.

Estimation: The final status of the application including total data size, average/maximum data size per scene, expected cost, etc.

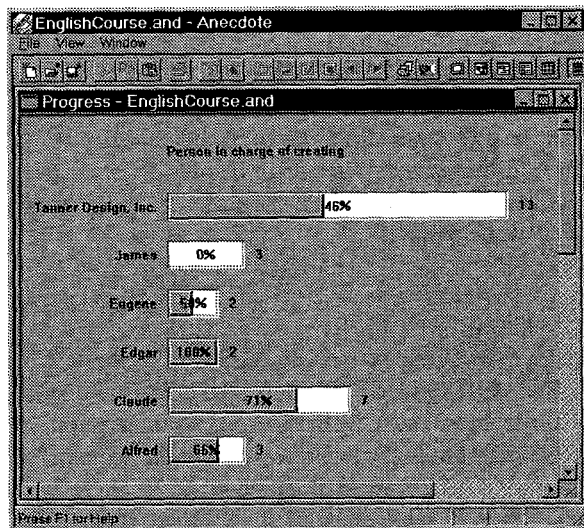


Figure 6: Progress Monitor

Scheduling: The progress of the development determined by the estimated scale of the final application, the current status, the start date and end date of the project.

While editing the scenario, authors can input user-defined

attributes, such as “person in charge of creating,” “due date” to each cast member. Progress Monitor can display the analysis, estimation and scheduling categorized by those attribute values (Figure 6).

4.4 Example authoring process with *Anecdote*

A typical sequence of multimedia title development using *Anecdote* is as follows:

In the Scene View, authors create slots with sketches and text annotations to describe the contents of the cast member (Figure 2). After completing one scene, authors can create a new scene and start editing that scene. If authors choose “New Scene” menu, a blank scene is created. If they choose “Copy Scene,” the slots in the last-edited scene are copied into the new scene. If authors select a slot before this operation, a link is automatically created from that slot in the last scene to this newly created scene. For instance, if authors want a button to appear when the audio playback ends, they should select the audio, choose “Copy Scene,” and add a button in the new scene. The event type of a link associated to an audio is set to “end of playback” by default. When authors switch to the Link View, they see

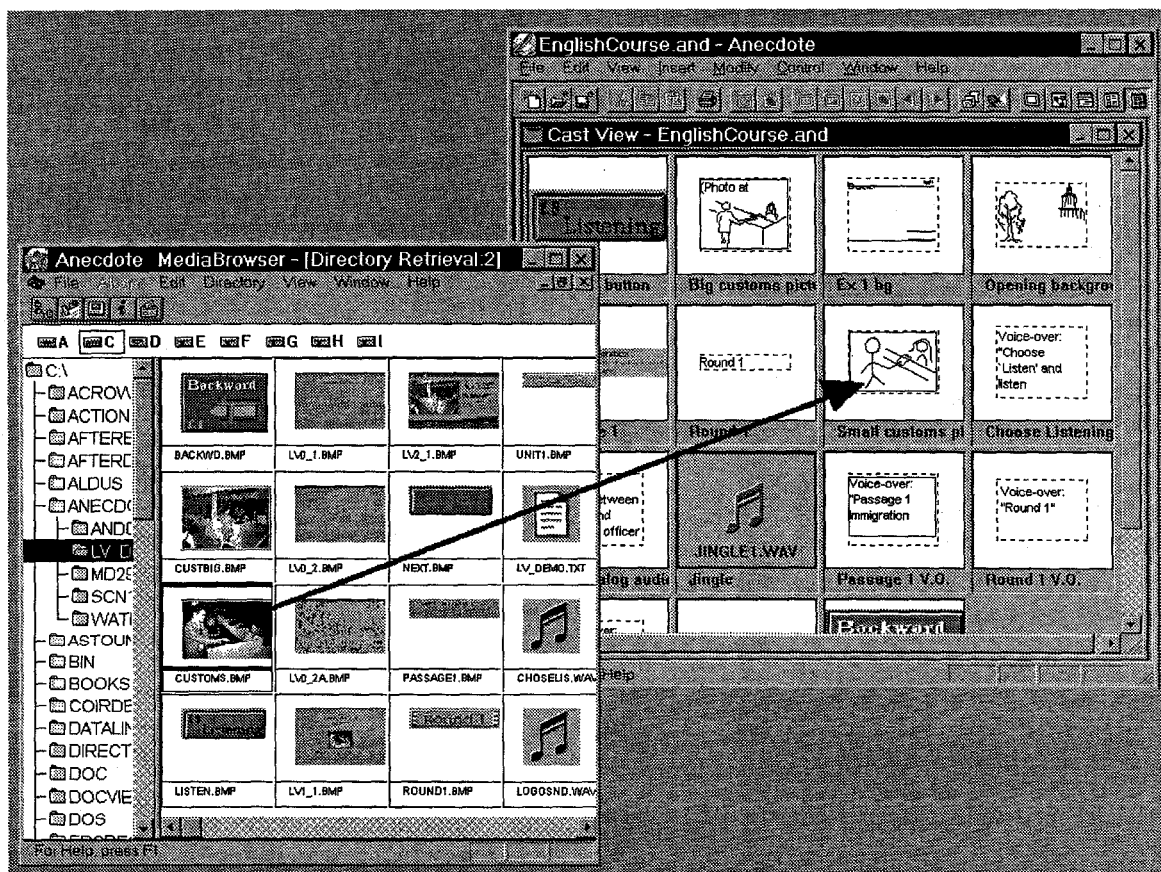


Figure 7: Media Browser and Cast View

The arrow is superimposed over the screen image to illustrate the drag & drop operation.

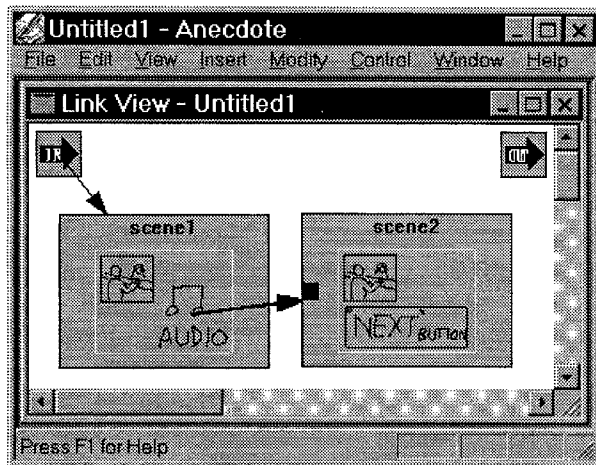


Figure 8: Example authoring process

the link created between the two scenes (Figure 8).

In the Link View, authors create additional links between the scenes (Figure 3). Authors can also create new scenes using “New Scene” and “Copy Scene” menus.

After some number of scenes have been created, authors select “Play” menu to execute the mock-up scenario. The scenario is executed using surrogate media. Authors can check the behavior of the application and make changes if necessary. Authors can also use the Progress Monitor to estimate the data size of the final application.

After completing the prototype scenario, authors can list all the cast members in the Cast View (Figure 7). The cast members with surrogate media indicate the media contents to be created. Authors might want to print out that list on paper and give it to other people in charge of creating the media contents. According to such specifications, media contents are created and delivered into the system.

Then, authors run a media contents management program to browse through the files. Figure 7 shows the

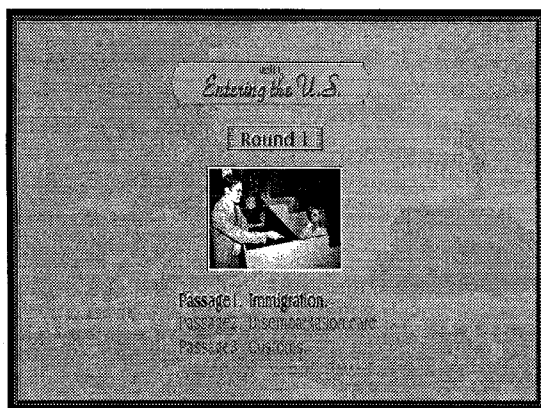


Figure 9: Scenario executed with media contents

MediaBrowser developed together with *Anecdote*. Authors can pick up a media file and drag & drop it into the corresponding cast member in the Cast View, Scene View or Timeline View. The change is immediately reflected in all other views.

When all surrogate media have been replaced with actual media contents and necessary readjustment to screen layout and display timing are made, the application becomes complete (Figure 9).

5 SUPPORT FOR VARIOUS AUTHORIZING STYLES

There are various styles of application design, including the typical authoring style described in the previous section. With *Anecdote*, authors can choose from different authoring styles depending on their preferences and types of applications. In the following subsections, we will explain how the editing views of *Anecdote* can help authors to design the scenario in their own styles.

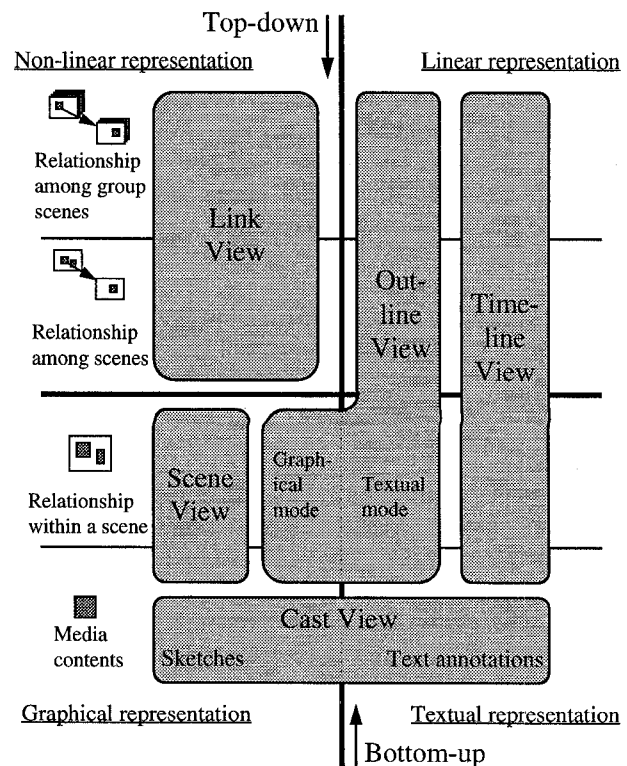


Figure 10: Scenario structure and editing views

5.1 Scenario structure and editing views

There are so-called “top-down” and “bottom-up” design strategies. In **Figure 10**, the four levels from top to bottom represent (1) the relationship among group scenes, (2) relationship among scenes, (3) relationship within a scene, and (4) contents of media objects². The top-down approach starts from the top level and proceeds downward. The bottom-up approach starts from the media contents level and goes upwards.

In the upper half of **Figure 10**, the left side represents the non-linear, hypermedia aspect of the scenario. *Anecdote* provides the Link View to edit this aspect of the scenario. The right side represents the linear aspect of the scenario defined by the timeline path. For this, the Outline View and the Timeline View are provided.

In the lower half of **Figure 10**, the left side represents the graphical aspect of the scene layout. The right side sees the content of a scene as a list of media objects displayed in a sequence. The Outline View has both the graphical mode and the textual mode to display these two aspects of the scenes.

With conventional authoring systems, authors must start the authoring process from the bottom of **Figure 10** and proceed upward through a narrow path. With *Anecdote*, authors can start from any design level of **Figure 10**, and proceed upward and downward in parallel, switching between the two-dimensional views (left) and linear views (right).

In next sections, we will have a look at the design processes using these different views and approaches.

5.2 Bottom-up vs. top-down

Bottom-up

The bottom-up authoring process starts in the Cast View. Authors can import existing media contents or input surrogate media to create cast members, and then drag & drop it into the Scene View and design the screen layout. This style of screen design is effective when the screen is designed with individual media objects and buttons.

In the Link View, authors can select more than one scenes and/or group scenes and then choose “Group” menu. The selected scenes and group scenes are moved into the newly created group scene. As shown in **Figure 11**, authors can select a scene that represents the group scene, and the contents of that scene is displayed in the group scene icon. The small squares on the sides of scene group icons and in the top-left and top-right corners of the window are called

² The design model of these four design levels is discussed in [11].

in-ports and *out-ports*. They are used to connect links between the lower- and upper-level scenes in the hierarchy.

In the Outline View, the hierarchical structure of scenes are displayed using indentations as in so-called “outline processor.” Authors can select scenes and choose “Group” the same way as in the Link View.

Top-down

In the Link View, authors can first design top-level framework using several surrogate scenes, each representing a subtree of the hierarchy to be created later. Then authors can pick up one of the surrogate scenes and select “Expand Scene” menu. A new group scene is created in place of that surrogate scene, and that surrogate scene is moved into the newly created scene group and registered as the scene representing the scene group.

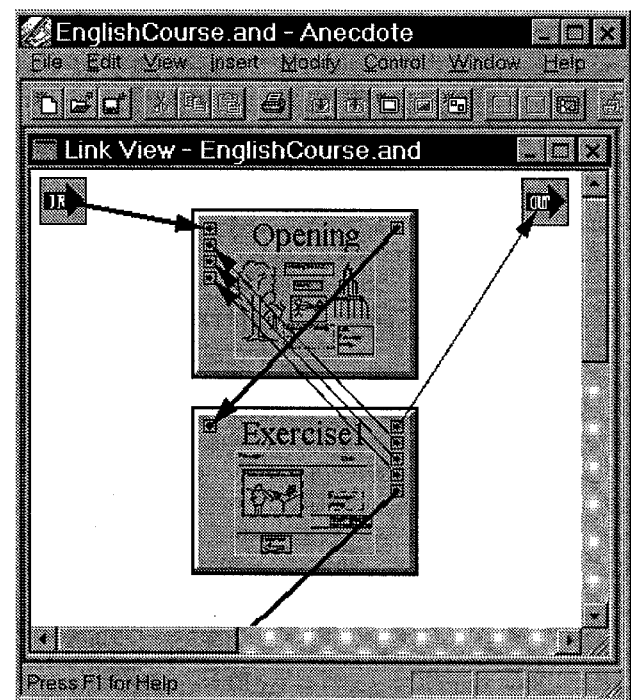


Figure 11: Group scenes

In the example of **Figure 3**, the basic screen designs are the same in the first four opening scenes, and the title logos are displayed one by one. This type of timed sequence can be created by following these steps:

1. In the Scene View, create the screen design of the surrogate scene by inputting all the sketches into the background of the scene.
2. In the Link View, choose “Expand Scene” to make a group scene with that scene inside.
3. In the Scene View, enclose some portion of the

sketches and select it, and then use “Create Slot” menu to define that portion as a new slot. The sketches in that portion move into the newly created slot.

4. In the Link View, copy the scene several times to make the timed sequence. In the Scene View, add or delete slots in each scene accordingly.
5. In the Link View, create the appropriate links to connect the scenes in the display order and create additional links for user interaction.

This process also can be done similarly using the Outline View.

5.3 Hypermedia vs. timeline

In the Timeline View and the Outline View, the scenes specified by timeline path are displayed in a sequence. (For editing convenience, the scenes not included in the timeline path are displayed at the end.) The timeline path is automatically defined while authors are creating new scenes, and can be easily changed in the Link View by clicking the links in order. The selected links are displayed as thick arrows in the Link View (**Figure 3**). While editing in the Scene View, Link View, Timeline View and Outline View, if authors select a slot before creating a new scene, a link is created to connect to the new scene, and the new scene is automatically added to the timeline path.

Using this feature, media synchronization can be realized. For example, if a video file already exists, authors can import it into the Timeline View. The images of video frames are displayed in the timeline of the video file. Then authors create several scenes so that the video timeline goes across those scenes, and then they incorporate the media to be synchronized into each scene. The video will be played

throughout those scenes synchronized with other media.

The data model of *Anecdote* is basically a state-transition diagram. When authors delete a slot from a copied scene, there is ambiguity as to if they want that slot to disappear when this scene is played, or they want it to be deleted from other scenes as well. The same problem arises when authors changed the position of a slot. The Timeline View is considered to be a matrix of scenes and cast members. Therefore, in the Timeline View, authors can shrink or extend the timeline of a cast member in order to explicitly specify in which scenes they want that cast member to appear. Similarly, before changing the position of a slot, authors can click and highlight some portion of the timeline to specify the scenes to which the layout change should reflect.

5.4 Graphical vs. textual

While “sketching” is an effective method for visually designing multimedia applications, not all scenario writers want to draw sketches. There are also cases where the screen design is relatively less important in the early-stage design. For instance, for an educational application in which text media and audio narration play a big role, textual scenario can be a better input format.

Anecdote's Outline View has *graphical mode* and *textual mode*. In textual mode, only scene names, cast names and textual annotations in each cast member are displayed. When “Differential Mode” menu is selected, only the slots (casts) that newly appear in the scene are displayed together with the slot that the link originated from (**Figure 12**).

It is possible for the authors to start the design using the textual mode of the Outline View. Authors can construct the sequence of scenes by typing text annotations for each media. The slots created in the textual mode are given default coordinates, i.e. on the left edge of the screen from top to bottom in order. Later, when authors switch to the Scene View, they can move those slots from their default coordinates to appropriate positions and sizes. *Anecdote's* capability of finding the scenes with unedited layout helps authors in this process.

6 SAMPLE APPLICATIONS

Several professional designers have created sample applications using *Anecdote* and evaluated its authoring design styles. **Figure 13** shows a screen image from the application titled “The Enchanted Forest.” In this example, each fish on the screen works as a button to go to a different scene. To create a screen image like this, the sketching capability and the top-down editing function of the Scene View (“Create Slot” menu) turned out to be very effective. **Figure 14** shows another application titled

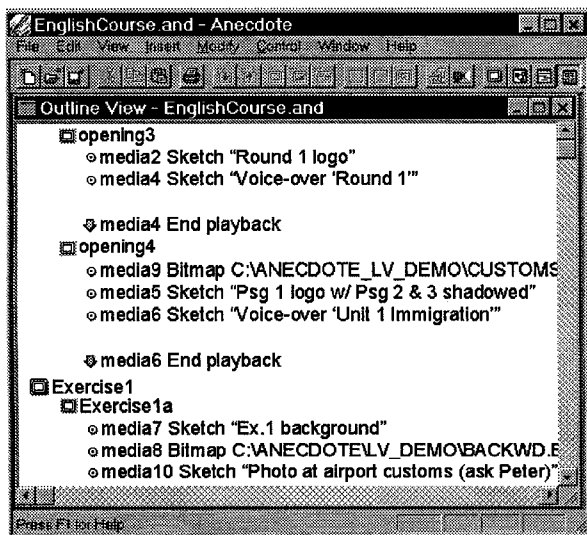


Figure 12: Outline View (Textual mode)

"Electronic Shopping Guide." This application includes as many as 77 scenes, and 25 group scenes were used to organize the scenario structure. Through the development of such applications, we got favorable comments from the designers and verified the usefulness of the authoring design principles of *Anecdote*.

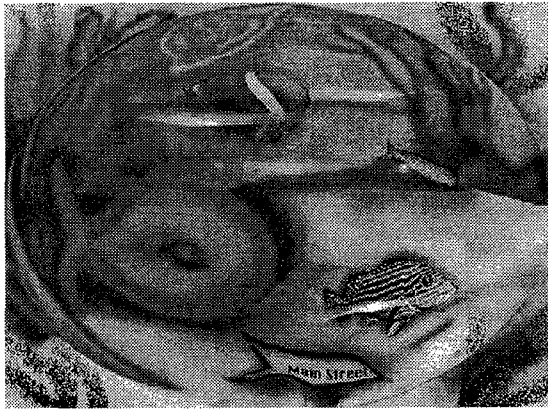


Figure 13: *The Enchanted Forest*



Figure 14: *The Electronic Shopping Guide*

authors to design the scenario using various authoring design styles and to transform the prototype into the final application smoothly.

We have implemented *Anecdote* and created sample applications to demonstrate the advantages of its authoring design principles. We hope the seamless authoring process described in this paper will benefit many multimedia designers and lead to the production of qualitative multimedia applications.

ACKNOWLEDGMENTS

We are grateful to our colleagues in C&C Research Laboratories, NEC USA, Inc. and Information Technology Research Laboratories, NEC Corporation, in particular, Kojiro Watanabe and Yosuke Takashima for their encouragement and fruitful comments on this work. We would also like to thank the designers at E.D.S. Design & Animation and NEC Design, Ltd. for their excellent design work and invaluable comments.

7 CONCLUSION

We have described the features of the *Anecdote* authoring system for the early-stage design and seamless development of multimedia applications. We have proposed the notion of surrogate media and surrogate scene that can be easily inputted to represent the content of the media data and a segment of the application structure, respectively. Surrogate media and surrogate scene are used in the place of the actual data to be created when the scenario is executed. They also constitute the specifications for creating the media contents and the scenes.

We have also proposed the timeline path that enables linear editing view of the hypermedia structure. We have described the various editing views of *Anecdote* that enable

REFERENCES

1. Asymetrix, *ToolBook User Manual (ver 3.0)*, 1994.
2. Kurlander, D. and Feiner, S., A Visual Language for Browsing, Undoing, and Redoing Graphical Interface Commands, in *Visual Languages and Visual Programming*, Shi-Kuo Chang, ed. Plenum Press, 1990, 257-275.
3. Landay, J.A. and Myers, B.A., Sketching Storyboards to Illustrate Interface Behaviors, in *Proceedings of CHI '96: Human Factors In Computing Systems*, 1996, 193-194.
4. Landay, J.A. and Myers, B.A., Interactive Sketching for the Early Stages of User Interface Design, in *Proceedings of CHI '95: Human Factors in Computing Systems*, 1995, 43-50.
5. Lewis, H., NoPumG: Creating Interactive Graphics with Spreadsheet Machinery, in *Visual Programming Environments*, IEEE Computer Society Press, 1990.
6. MacNeil, R., Generating Multimedia Presentations Automatically using TYRO, the Constraint, Case-Based Designer's Apprentice, in *Proceedings of the IEEE 1991 Workshop on Visual Languages*, 1991
7. Macromedia, *Using Director 5.0*, 1994.
8. Modugno, F. and Myers, B.A., A State-Based Visual Language for a Demonstrational Visual Shell, in *Proceedings of IEEE/CS Symposium on Visual Languages*, 1994.
9. Nanard, J. and Nanard, M., Hypertext Design Environments and the Hypertext Design Process, in *Communications of the ACM* 38, 8 (Aug 1995), 49-56.
10. Ogawa, R., Harada, K. and Kaneko, A., Scenario-based Hypermedia: A Model and a System, in A. Rizk, N. Streitz and J. Andre, eds., *Hypertext: Concepts, Systems, and Applications*, Cambridge University Press, 1990, 38-51.
11. Ogawa, R., Tanaka, E., Taguchi, D. and Harada, K., Design Strategies for Scenario-based Hypermedia: Description of its Structure, Dynamics, and Style, in *Proceedings of the ACM Conference on Hypertext (ECHO '92)*, 1992, 71-80.
12. Wellner, D., Statemaster: A UIMS based on Statecharts for Prototyping and Target Implementation, in *Proceedings of CHI '89*, 1989, 177-182.
13. Zielinski, R.S., *Using Macromedia Authorware 3.5*, QUE Corp., 1996.