AESOP: AN OUTLINE-ORIENTED AUTHORING SYSTEM

Takeshi Shimizu, Stephen W. Smoliar, and John Boreczky

FX Palo Alto Laboratory, Inc.; 3400 Hillview Ave., Bldg. 4; Palo Alto, CA 94304
{shimizu,smoliar,johnb}@pal.xerox.com

Abstract

Because a hypermedia document is more complex than conventional text, it requires preparation with respect to two key aspects. First, the author begins to develop a "vision" of the document—usually based on some outline-level description of his objectives. At the same time, as this outline is being developed, the author begins to extract useful segments from his resource materials and prepares his first version of the logic of a system of hyperlinks among those segments. In this paper we present a system named "Authoring Environment for the deSktOP" (AESOP) with two different types of "outlining" tools to handle these aspects. Planning the "vision" consists in defining a "logical" tree structure of the document. The plan for the link structure is based on a primitive unit called the view area*, and AESOP provides a construct named* Bento-Box *for creating and manipulating view areas. Authors specify spatial and temporal layout within a single Bento-Box and define hyperlinks among the Bento-Boxes.*

1. Introduction

A major objective in the design of authoring systems has been the grounding of the system in a suitable cognitive model for how writing takes place. One of the earliest efforts in this direction was a project at the University of North Carolina that developed the Writing Environment, a system to support writing verbal text documents [14]. The system provided four workspaces, called the *network mode*, the *tree mode*, the *editor mode*, and the *text mode*, that were based on a cognitive model for authoring that reflected the need to organize resources before writing and to structure outlines according to both logical and presentational criteria. SEPIA [17] is based on a similar cognitive model and also provides four workspaces, this time called the *content space*, the *rhetorical space*, the *planning space*, and the *argumentation space*. However, SEPIA was designed to handle hypermedia, rather than just verbal text. In SEPIA's case the planning space is concerned with logical structure, while the rhetorical space addresses presentational criteria. The Instructional Design Environment (IDE) took a similar approach in providing tools to support the collection, logical organization, and presentation of course materials [12]. The CMIF authoring environment [10] allows structure-based composition of hypermedia documents. The system provides two different views:

1) A Hierarchy view to edit the hierarchy of components in a hypermedia document.
2) A Channel view to specify synchronization among the components.

The Hierarchy view is used for a *top-down* construction of the document and thus deals with information concerned with logical structure. The Channel view is used for a *bottom-up* construction of the details which have more to do with presentational criteria.

Unfortunately, neither the CMIF authoring environment, the Writing Environment, SEPIA, nor IDE supports an explicit representation of relationships between the logical and presentational organizations of the document being authored. Thus SEPIA provides no assistance when the writer wishes to move from the planning space to the rhetorical space. Because the workspaces are, for all intents and purposes, independent, this also means that neither system provides any visualization of the author's overall progress in working on a document (for example giving the author some idea of how much of the document has been completed or how a change in the logical structure may have an impact on the presentational structure).

Using templates is a common approach to maintaining consistency of the content during hypermedia authoring [2]. Also it makes the authoring easier because authors do not need to worry about layouts every time new content is created. IDE provides templates to construct individual nodes with pre-defined *constrained* links [12]. PageJokey [8] also provides templates for making hypermedia documents based on a *page* metaphor. The PageJokey system allows users to specify the temporal presentation of the pages and the spatial layout within pages using templates. Neither IDE nor PageJokey provides different types of templates for both logical and presentational organization of the document. Finally, no existing system provides very much support for navigation by the user, either within a workspace or among the multiple workspaces. All of these difficulties can result in users, particularly novices, becoming easily disoriented in the course of an authoring task.

We now present a system named "Authoring Environment for the deSktOP" (AESOP). Like the Writing Environment the design of this system is based

1060-3425/98 $10.00 (C) 1998 IEEE

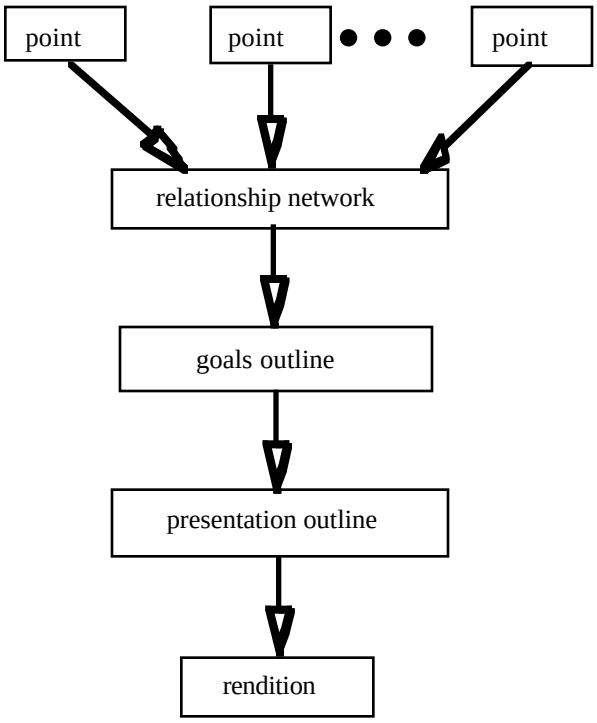

Figure 1: Model of reading and writing.

on a cognitive model of writing; but it has also been designed to support more facile movement among the tasks upon which the model is based (Section 2). The AESOP user sorts out concerns for logical and presentational criteria through user interfaces for two different types of outlining. The need for this explicit division of outlining is discussed in Section 3; and the types are discussed in greater detail in Sections 4 and 5, respectively. In addition, AESOP has a meta-level user interface, discussed in Section 6, based on a "Kitchen" metaphor which helps authors shift attention from one task to another in the workspace for hypermedia authoring. The metaphor provides an intuitive "navigation sense" for what tasks should be performed while creating hypermedia documents.

2. Design Goals

The model of writing on which AESOP is based is illustrated in Figure 1. In addition to being basically compatible with the model behind the Writing Environment [14], this model is also consistent with the model of strategic discourse processing developed by Teun van Dijk and Walter Kintsch [6]. The Writing Environment model is based in eight *cognitive modes*, while the model of van Dijk and Kintsch consists of six *strategies*. Let us now review the stages of the model in Figure 1:

1) One collects individual units of content, which we call *points*, not only when one is preparing to write (in the cognitive model of *exploration*) but also when one is trying to understand what one is reading; and the resulting collection is the product of what van Dijk and Kintsch call *propositional strategies*.

2) As points are accumulated, they are organized into a *relationship network* (another instance of the cognitive mode of exploration), following what van Dijk and Kintsch call *local coherence strategies*.

3) The resulting organization can then assist either the writer or the reader in identifying the goals of the text—what the text is trying to communicate. This understanding of goals involves the cognitive mode of *situational analysis* in the Writing Environment and arises from what van Dijk and Kintsch call *macrostrategies*. In our model, it results in a *goals outline* that provides a hierarchical representation of the goal structure.

4) How this plan is actually realized in a document, however, is based on the cognitive mode of *organization* in the Writing Environment and what van Dijk and Kintsch call *schematic strategies*; our model has postulated a *presentation outline* to represent the structure of this realization.

5) Finally the *rendition* of the document, itself, is a relatively straightforward matter of "fleshing out" this skeleton (the cognitive mode of *writing* in the Writing Environment) according to what van Dijk and Kintsch call *production strategies*.

In summary the most important feature of our model is that every box in Figure 1 corresponds to an artifact that can be associated with the tasks of both writing and reading; and one of our primary design goals is to support the construction of those artifacts in the course of either of those tasks.

As was observed in Section 1, both the Writing Environment and SEPIA deal with the different cognitive modes involved in writing by providing the writer with separate workspaces. Unfortunately, these workspaces are not only *physically* separate (each operating in its own window) but also *logically* separate. There is little support for the "propagation" of activities taking place in one workspace into the "state" of the others. This obliges the writer to keep the multiple workspaces reliably consistent with each other.

Nevertheless, AESOP has identified (and begun to implement) several techniques that facilitate the writer keeping up with this demand, providing a seamless transition among the stages of the model in Figure 1:

1) To the extent that it is possible, the *importation* of media data is supported at more than the surface level, providing structural information along with the media objects.

2) All media objects are maintained in a database *indexed according to properties of the specific media*, as opposed to text labels [16].

3) AESOP provides a *Document Prototype*, which supports a template-based approach to the construction of goals outlines based on a theory of document types.

4) Construction of the presentation outline is based on a primitive unit called the *view area*, and AESOP provides suitable visual aids for the construction and management of view areas.

5) The major problem in planning a document is *consistency between the goals and presentation outlines*; AESOP provides visual cues to assist the writer in maintaining this consistency.

6) AESOP provides means for browsing the document, viewed as a whole, in order for the writer to maintain a sense of *progress*: How much has been completed, and how much work remains?

7) AESOP provides an intuitive user interface for *navigating among its different workspaces*.

In this paper we shall concentrate our attention on techniques 3), 4), and 7).

3. Two Types of Outlining

3.1 Why Two Types?

One of the most important lessons we have learned from our experiences in authoring hypermedia documents [15] is that outlining is as important in hypermedia as it is in conventional writing. As a matter of fact, we would argue that outlining is even *more* important, because we feel our documents have provided support for the model of reading and writing illustrated in Figure 1, which postulates that proper authoring requires more than one outline. While the goals outline provides the general plan for the *content* of the document, it is still necessary to plan out how the hypermedia document, itself, will be organized; and this latter plan is represented by the presentation outline. In the text medium a table of contents tends to serve reasonably well as a presentation outline, but when other media are involved the principles behind a table of contents cannot be generalized. Consequently, storyboards generally serve as presentation outlines for films; and sketches or cartoons tend to be used as presentation outlines for images. When hypermedia links are also involved, the outline must accommodate not only the *appearance* of the document but also its *affordances for interaction*. Thus, our intuitions for outlining verbal text do not carry over to hypermedia in a straightforward manner.

3.2 Serving Both Author and Reader

However, these two types of outlines are not only valuable in the course of writing. As the model of van Dijk and Kintsch [6] has demonstrated, such outlines are constructed mentally in the course of reading verbal text. Because hypermedia tends to confront the reader with far more complexity, however, the mental construction of such outlines can be far more difficult. Instead, the reader will frequently benefit from having these outlines already constructed as part of the document. In this capacity they serve as "road maps" for the reader. The presentation outline provides the reader with an overview of the entire document, which will generally entail some sense of the amount of time and effort that will be required for reading it. Furthermore, if the document is based on a rich and complex structure of links, the presentation outline may suggest how the reader may want to traverse these links, making the concept of a "road map" a bit more than metaphorical. This is a service that cannot be readily provided by a text outline, and printed books that implicitly implement such linked structures are often rather poorly served by such constructs as tables of contents or indexes. At the same time the goals outline provides a map of what the document is trying to say. If the reader does not have the time to devote to the entire document, the goals map should indicate where attention should be focused in order to glean that particular information that the reader actually needs.

4. The Goals Outline

4.1 What It Is

As was observed in Section 2, a goals outline is basically a hierarchical structure that represents what the document is trying to communicate. As an example we authored a hypermedia report of a group meeting at our laboratory that discussed the value of logic as a technology [15]. The goals outline for this document was the following:

I. What constitutes the technology of logic?
II. Past successes
III. Hypothesized successes
    A. Can logic be used to express common sense?
        1. Refutations
        2. Rebuttals
    B. Can logic fulfill software technology needs?
        1. Refutations
        2. Rebuttals

This outline could then serve as a logical framework for all the material that had been collected and implemented as points for the documentation of this discussion.

4.2 "Logical" Templates for Document Types

How can we facilitate the construction of goals outlines? We must begin by recognizing that not all documents are going to have goals outlines structured according to the same principles. The highest-level goals of documents can differ in very significant ways; and we argue that determining a goal at the top level can have an impact on how the lower levels will be structured. For this reason we have decided to classify these top-level goals according to the *text types* of narratology [4]. In a previous paper [15] we demonstrated that the theory of text types could be extended to accommodate hypermedia documents. We shall now examine in greater detail how a text type can determine the structure of a goals outline. We shall examine the three text types that were discussed in that paper: Description, Argument, and Narrative.

4.2.1 Description. In our previous work [15] we tried to approach Description as dealing with some *object* (which need not be concrete) that needs to be described. Then, from a knowledge representation point of view, we can deal with the task of description as one of divide-and-conquer: decompose the object into some set of component parts, describe each part (recursively as a new description task), and describe how they are assembled to form the whole. This may be taken as a strategy for writing documents of the Description type. Admittedly, this strategy does not do very much justice to many of the literary examples of Description [4]; but also it does not necessarily preclude them.

According to this strategy, then, we need to construct a relationship network whose primary job will be to represent part-whole relationships. The goals outline will be very intimately related to this relationship network. Indeed, probably the most important thing about the goals outline is that the writer needs to decide *how much* of the relationship network is actually going to go into the document. Also, the goals outline needs to address when the document is going to discuss how the parts combine to form the whole, rather than simply enumerating the parts.

4.2.2 Argument. For documents of the Argument type, we may turn to the approach taken by the SEPIA system [17]. SEPIA drew upon a theoretical investigation of argument undertaken by Stephen Toulmin [18]. Toulmin classified all of the propositional statements that may be invoked in an argument into six categories: Claim, Datum, Warrant, Rebuttal, Qualifier, and Backing. This approach can then form the basis for a relationship network that classifies points according to these categories. Again, this does not necessarily do justice to all the literary subtleties of argument [4]; but, also again, it need not preclude them. Furthermore, we may treat Toulmin's diagrammatic representation of the relationships among these categories as a goals outline, which is basically what SEPIA has done.

4.2.3 Narrative. Thus far we have had our least success in trying to deal with documents of the Narrative type [15]. However, in his book *Story and Discourse* Seymour Chatman [3] has done for narrative something very similar to what Toulmin did in his analysis of argument: He has proposed a set of categories that can be used to classify the points that go into making a Narrative. These categories may be outlined as follows:

I. Story (Content)
  - A. Events
    - 1. Actions
    - 2. Happenings
  - B. Existents
    - 1. Characters
    - 2. Settings
  - C. People, things, etc., as preprocessed by the author's cultural codes

II. Discourse (Expression)
  - A. Structure of narrative transmission
  - B. Manifestation
    - 1. Verbal
    - 2. Cinematic
    - 3. Pantomimic
    - 4. etc.

This may not work in its entirety. In particular a primary distinction between the goals and presentation outlines is that, in the terminology of the above outline, the former is concerned with content, while the latter is concerned with expression. Thus, we may wish to restrict out attention to the Story portion of the above structure as a basis for constructing a category-based relationship network.

If we focus on the Story categories, then we have a way of proceeding from the relationship network to a goals outline. We can do this by adapting an approach proposed by Umberto Eco [7] in *The Role of the Reader*, that was, in turn, adapted from the *Poetics* of Aristotle [1]. According to this approach, the root of the goals outline involves the statement of an agent (human or otherwise), an initial state, and a final state (which is not necessarily a permanent one). This node may then be refined into a series of subnodes, each of which represents a time-oriented change, marking out the logical path from the initial state to the final one. Each of these changes may then be elaborated with respect to agent, initial state, and final state, as well as causal relationships to other changes. In all fairness, however, this work is still very preliminary. For documents of both the Description and Argument types, we have investigated examples upon which we have drawn for structuring appropriate templates. We are still working on comparable examples for Narrative documents, so we are not yet to the point of being able to design the corresponding templates.

4.3 Implementing Goals Outlines

For each of the above text types, the strategy for writing a document may be encapsulated as a template that we call a *Document Prototype*. Taking an object-oriented approach, these Prototypes are implemented as classes and maintained in a class hierarchy that may be updated by the user. Instances of these classes are used to construct goals outlines and may be combined to implement the embedding of different text types. The construction of the actual goals outline then involves the management of three types of links:

1) *Mandatory links* are defined by the class and must be included in all instances.

2) *Optional links* are defined by the class but may be removed by the user.

3) *User-defined links* are not defined by the class but are added by the user.

4.4 Combining Document Types

An important property of text types is that they may be *embedded*. For example in literature it is frequently the case that Descriptions are embedded in a text that is, at the top level, Narrative [4]. Thus, when a goals outline is constructed, any interior node may be realized as an instantiation of another Document Prototype. Indeed the "depth" of part-whole relationships in the goals outline for a Description is basically implemented by recursively instantiating the Document Prototype class for Description for some, if not all, of the interior nodes of the goals outline.

5. The Presentation Outline

As was observed in Section 2, the presentation outline is based on the concept of a *view area*. "Spacer" objects represent appearance in a single view area and can account

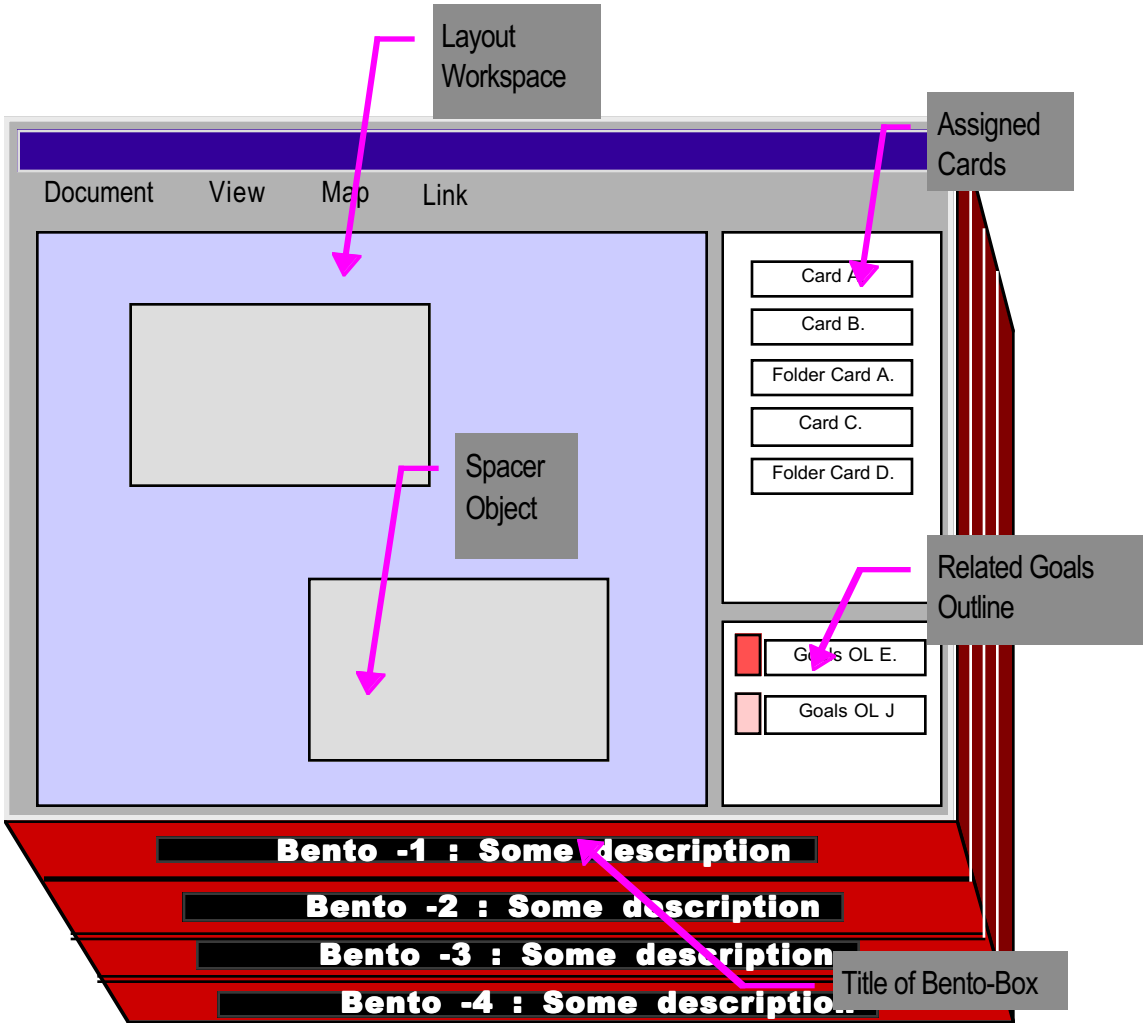

Figure 2: User interface for the Bento-Box.

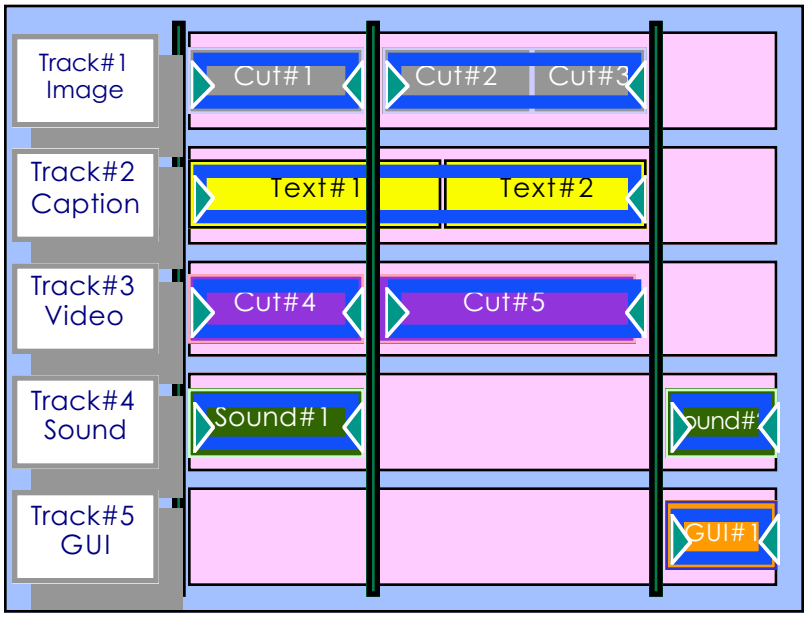

Figure 3: The Temporal View of the Layout Workspace

for both spatial and temporal structuring. Link presentation is then represented by navigation paths among multiple view areas. Unlike many hypermedia systems, such as the World Wide Web, AESOP supports documents that allow the reader to examine more than one view area at a time. The *Bento-Box* is the primitive unit for building presentation outlines. (The name "Bento" is also used in HyTime [11] as *sbento*, which is a container object for multiple-entity data; we are using the Bento-Box to define a unit for both data structure and visual interface.) Each Bento-Box plays the role of a paragraph in a hypermedia document [13]. The user designs the Bento-Box to identify "spacer" objects and provide their spatial and temporal specifications. Links may then be specified among both "spacer" objects and Bento-Boxes. In addition to representing such general linking information, Bento-Boxes may be "stacked" to represent that a set of view areas should be presented as a temporal sequence.

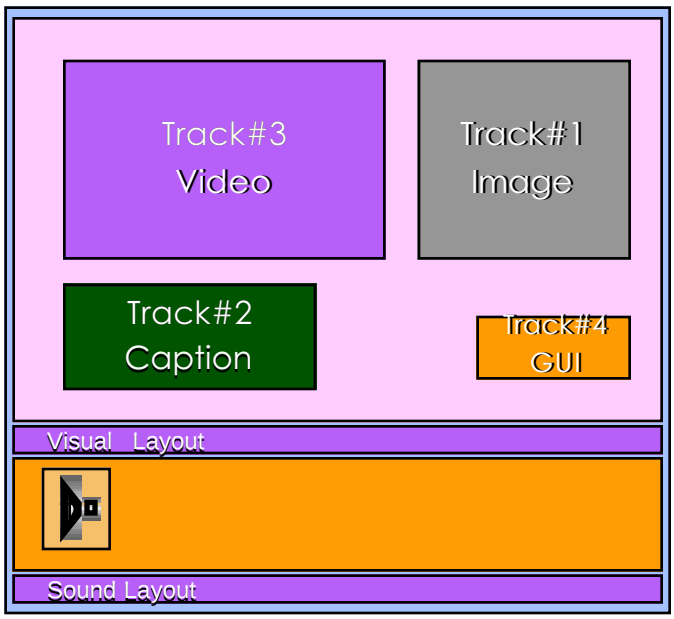

Figure 4: The Spatial View of the Layout Workspace

Figure 2 illustrates the user interface through which a stack of Bento-Boxes is constructed. The interface for an individual Bento-Box consists of four areas:

I. *Layout Workspace*: This is where the structure of the presentation outline is specified through "spacer" objects; there are three modes for viewing this area:
   - A. The Temporal View specifies time and synchronization among "spacer" objects (Figure 3).
   - B. The Spatial View specifies locations of "spacer" objects (Figure 4).
   - C. The Logical View specifies logical dependencies among "spacer" objects (Figure 5).

II. *Assigned Cards*: Architecturally, AESOP is based on NoteCards [9]. Consequently, all points are represented as Card objects. This area is used to show which points have been assigned to this Bento-Box.

III. *Related Goals Outline*: After the user has assigned content to the "spacer" objects, this

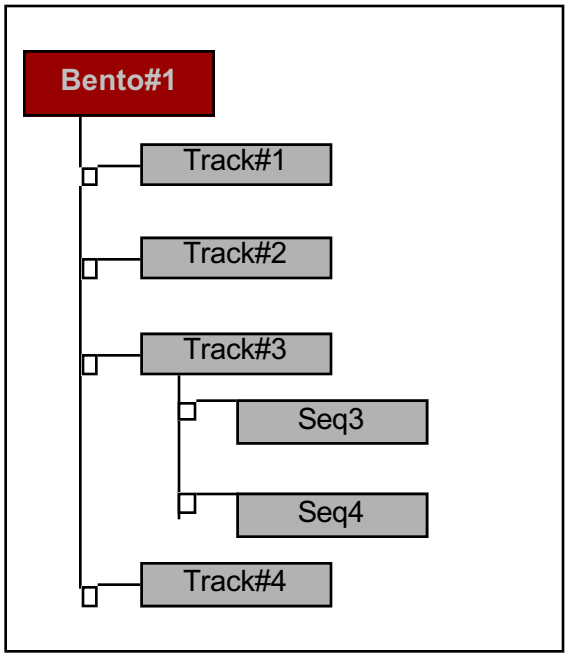

Figure 5: The Logical View of the Layout Workspace

area shows icons for the nodes in the goals outline that are associated with those "spacer" objects.

IV. *Title*: This serves to label Bento-Boxes piled in a stack. In Figure 2 there are four Bento-Boxes in the stack. The user clicks the Title to specify which Bento-Box is to be edited.

"Spacer" objects do not require content while the user is working with them in the Layout Workspace. Content need only be assigned once the user associates a "spacer" object with an element in the goals outline. The manner in which content is assigned is determined by the type of the "spacer" object:

1) Paste type: The source contents are "pasted" onto the "spacer" object.

2) Traversal type: The "spacer" object serves as a "button" that conducts the reader to the site of the source contents.

3) Composite type: The "spacer" object serves as a "menu" from which the user selects the source contents to be observed; this "menu" can be a list box, a set of radio buttons, a set of check boxes, or a dialogue box.

Figure 6 shows an example of how a stack of Bento-Boxes represents a presentation. In this case three consecutive slides are represented as a stack of three Bento-Boxes. Figure 7 shows how a basic slide presentation may be made more elaborate to accommodate hypertext links.

As is the case with goals outlines, the construction of presentation outlines may be supported by a library of templates. These templates may be used to specify

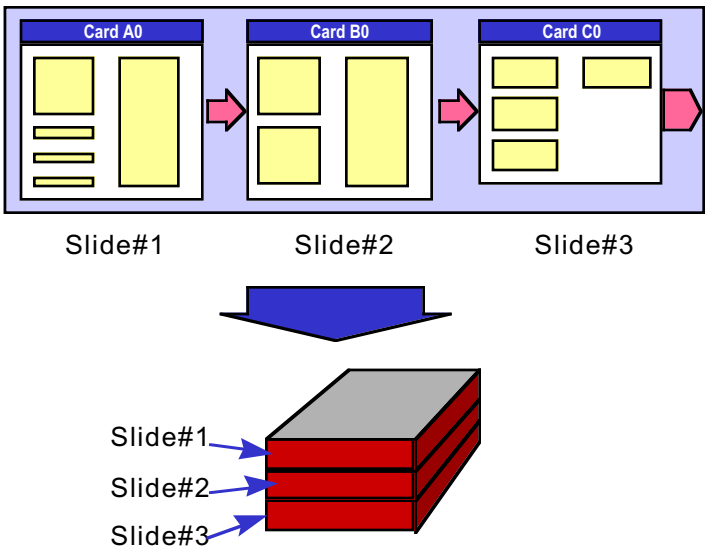

Figure 6: Bento-Boxes for a slide sequence.

structures for the different views in the Layout Workspace or for the linked organization of a stack of Bento-Boxes. *How* a library of templates should be organized and indexed is still under investigation.

6. Supporting Authoring Through a Meta-GUI

One of the biggest problems facing authors of hypermedia documents is the complexity of the material with which they are working. The danger of a reader getting "lost in hyperspace" has been frequently acknowledged [5]; but it is clear from the number of types of information, enumerated in Figure 1, with which the author must work that the author faces similar problems of "getting lost." Multiple workspaces are clearly necessary, but how can the author maintain a clear sense of which workspace he should be giving his attention to at any stage in the preparation of his document?

Negotiating complex tasks that require using multiple workspaces is nothing new. We probably all have at least one such task with which we are familiar. In AESOP we have been exploring how such familiarity may be exploited through the construction of an appropriate meta-level interface that constructs a metaphor for such a familiar task. This work is still very preliminary, but an example of such an interface is illustrated in Figure 8. We see here how the image of a kitchen works as a meta-level graphical user interface (GUI). In this case the "Recipe" on the cork board is used as an interface for the goals outline; and the cutting board and Bento-Boxes (now in the culinary context) are used as interfaces for the presentation outline. In addition the refrigerator and

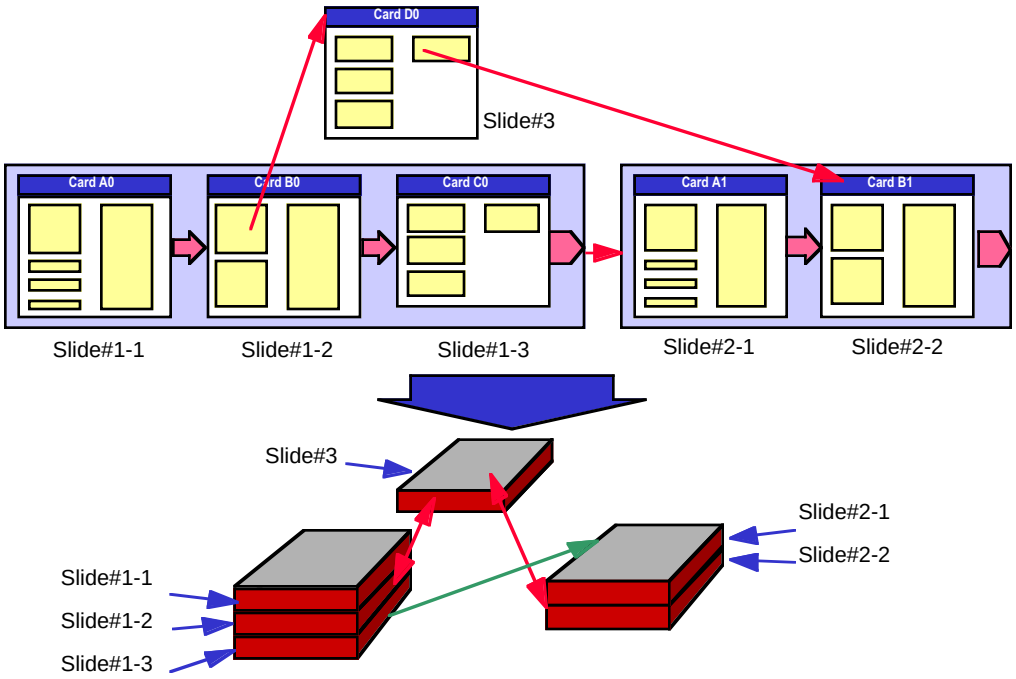

Figure 7: Hyperlinked Bento-Boxes.

cabinets are used as storage spaces and organizers for media resources.

7. Conclusion

Authoring hypermedia is not easy, but that does not mean it should be unmanageable. AESOP has been designed with the management of the complexity of authoring as its highest priority. On the one hand it draws upon results from cognitive sciences concerned with how we think about both writing and reading. At the same time it provides for a meta-level GUI through which we may engage other familiar tasks in a metaphorical capacity.

8. Acknowledgments

It would be very difficult to list all the participants who helped contribute to the ideas we have now documented. However, we feel it is important to thank those who have contributed to the implementation of AESOP: Eiji Ishida, Bee Liew, and David Beitzel. We also wish to thank Seymour Chatman, of the University of California at Berkeley, for hearing us out as we ventured into territory far more familiar to him than it was to us and helping us out with many valuable suggestions and guideposts. Finally, we wish to thank James Baker for not only his administrative support but also the active role he has taken in preparing several of his documents using our system.

9. Bibliography

1) Aristotle, "Poetics", In J. Barnes (ed.), *The Complete Works of Aristotle: The Revised Oxford Translation*, Princeton University Press, Princeton, 1984, pp. 2316–2340.

2) Catlin, K. S., Garrett, L. N., and Launhardt, J. A. "Hypermedia Templates: An Author's Tool", *Hypertext '91 Proceedings* (December 1991), pp. 147–160.

3) Chatman, S. *Story and Discourse: Narrative Structure in Fiction and Film*, Cornell University Press, Ithaca, 1978.

4) Chatman, S. *Coming to Terms: The Rhetoric of Narrative in Fiction and Film*, Cornell University Press, Ithaca, 1990.

5) Conklin, J. "Hypertext: An Introduction and Survey", *Computer*, Vol. 20, No. 9 (1987), pp. 17–41.

6) van Dijk, T. A., and Kintsch, W. *Strategies of Discourse Comprehension*, Academic Press, San Diego, 1983.

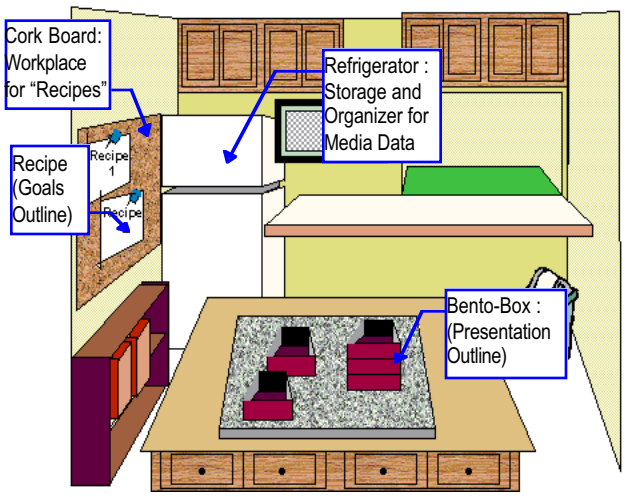

Figure 8: Using the image of a kitchen as a meta-level interface.

7) Eco, U. "The Role of the Reader", In *The Role of the Reader: Explorations in the Semiotics of Texts*, Indiana University Press, Bloomington, 1984, pp. 3–43.

8) Fraïssé, S., Nanard, J., and Nanard, M. "Generating Hypermedia from Specifications by Sketching Multimedia Templates", *ACM Multimedia 96 Proceedings* (November 1996), pp. 353–363.

9) Halasz, F., Moran, T., and Trigg, R. "NoteCards in a Nutshell", *ACM CHI+GI Conference Proceedings* (1987), pp. 45–52.

10) Hardman, L., van Rossum, G., and Bulterman, D. C. A. "Structured Multimedia Authoring", *ACM Multimedia 93 Proceedings* (August 1993), pp. 283–289.

11) ISO/IEC 10744, Hypermedia/Time-based Structuring Language (HyTime), 1992

12) Russell, D. M. The Instructional Design Environment: Six Years Before the Mast with IDE, presentation at Xerox PARC (July 1990).

13) Shimizu, T., Nakamura, O., and Kiyoki, Y. "Multimedia Document System for Temporal and Spatial Structuring", In P. Fraïssé *et al.* (eds.), *Hypermedia Design*, Springer, London, 1996, pp. 39–58.

14) Smith, J. B., and Lansman, M. A Cognitive Basis for a Computer Writing Environment, TR87-032, Department of Computer Science, University of North Carolina at Chapel Hill (June 1988).

15) Smoliar, S. W., and Baker, J. D. "Text Types in Hypermedia", *Proceedings of the Thirtieth Annual Hawaii International Conference on System Sciences* (January 1997), Vol. VI, pp. 68–77.

16) Smoliar, S. W., *et al.* "Multimedia Search: An Authoring Perspective", *Image Databases and Multi-Media Search Proceedings* (August 1996), pp. 1–8.

17) Streitz, N., *et al.* "SEPIA: A Cooperative Hypermedia Authoring Environment", In R. Rada (ed.), *Groupware and Authoring*, Academic Press, London, 1996, pp. 241–264.

18) Toulmin, S. E. *The Uses of Argument*, Cambridge University Press, New York, 1964.