Appeared in the Proceedings of International Conference on Future Software Technology (ISFST) 2001, Software Engineers Associates, ZhengZhou, China, pp. 246-251, November, 2001.

ARTWare: A Component Library for Building Domain-Oriented Authoring Environments

Yasuhiro Yamamoto [1]

[1] Graduate School of Information Science, NAIST
8916-5, Takayama, Ikoma, 630-0101, Nara
yxy@is.aist-nara.ac.jp

Kumiyo Nakakoji [1,2,3]

[2] Intelligent Cooperation and Control Group, PREST, JST
2-2-11, Taruoka, Miyagino, Sendai, 983-0852, Miyagi
kumiyo@is.aist-nara.ac.jp

Atsushi Aoki [3]

[3] SRA Key Technology Laboratory, Inc.
3-12, Yotsuya, Shinjuku, 160-0004, Tokyo
aoki@sra.co.jp

ABSTRACT

Building interactive systems using multimedia objects requires a wide range of knowledge including knowledge about 3D objects, graphics, movies, sound, and how to design user interface. We have built ARTWare, a library of components for building domain-oriented multimedia authoring environments. ARTWare consists of interactive components, such as various types of movie viewers and 3D spaces for positioning multimedia objects. We argue that in order for such a library to be useful it has to provide components that are (1) well designed based on theories of Human-Computer Interaction and (2) well integrated by synchronizing behavior of objects across different components. Interactive multimedia systems using ARTWare illustrate how ARTWare helps the development of interactive multimedia application systems in different domains.

KEYWORDS

Human-computer interaction, middleware for domain-oriented multimedia authoring environments, movie viewers, externalization, spatial positioning

INTRODUCTION

It has become common to use multimedia objects on computer systems. A user can watch movies and model 3D objects on a regular personal computer without any specialized equipment. A number of commercial packages as well as shareware and freeware are available for viewing, editing, storing, and sharing multimedia objects.

However, if users have more specialized purposes, for instance, a doctor wants to analyze his/her patients' X-ray data or a psychologist wants to analyze videotaped subjects' data, existing multimedia application systems do not always fit to their needs.

We have built ARTWare, a library of components for building domain-oriented multimedia authoring environments, for instance, a system for empirical video analysis for usability analysts. ARTWare is for those who are in need to use computers to perform their tasks by using movies or 3D shapes, and for those who need to build such systems.

The components of ARTWare are software tools each of which provides functions and a user interface; each tool can be used as an independent system. They are generally at the higher levels of abstraction than most of software component libraries, which often do not provide user interfaces and cannot be "used" by end-users by themselves. ARTWare components can communicate with each other; data and controls are shared among the tools.

We have built domain-oriented authoring environments by instantiating components of ARTWare in different types of tasks. They illustrate how ARTWare can be used to help the development of domain-oriented multimedia authoring environments.

Appeared in the Proceedings of International Conference on Future Software Technology (ISFST) 2001, Software Engineers Associates, ZhengZhou, China, pp. 246-251, November, 2001.

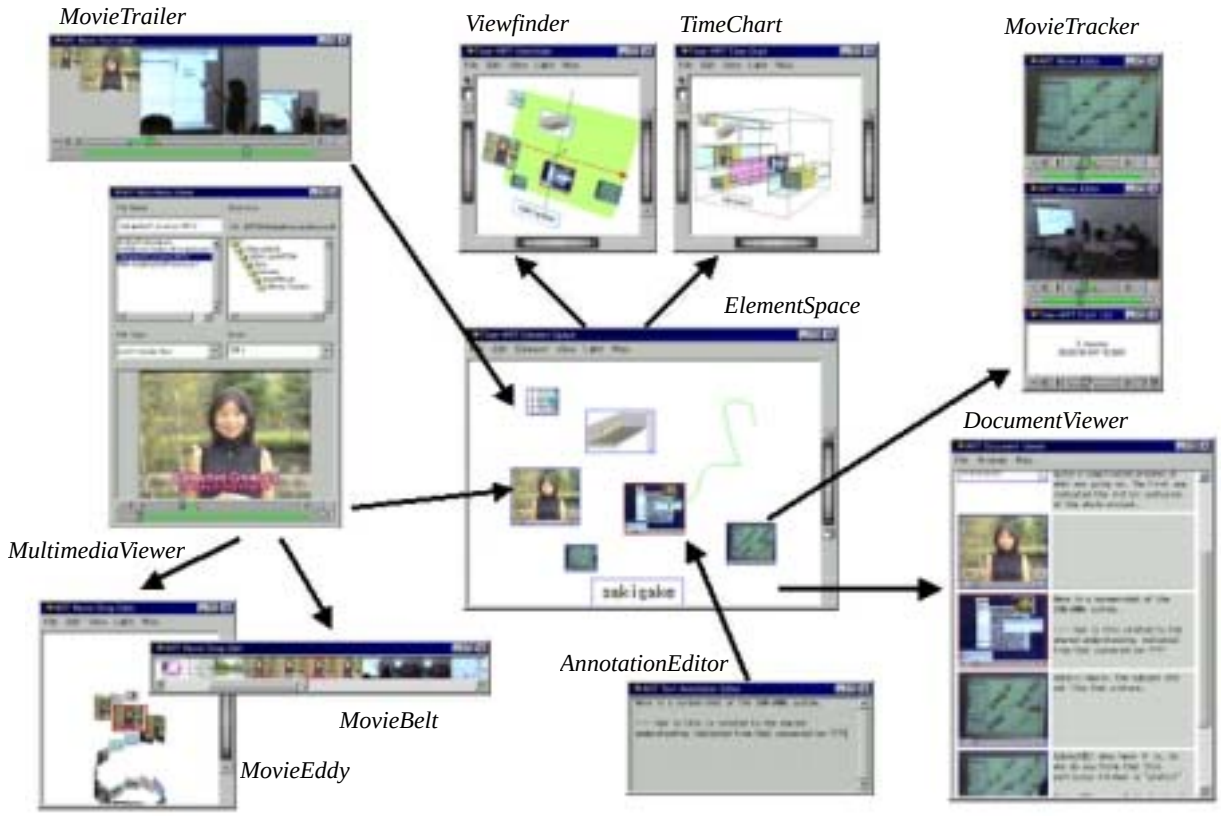

Figure 1: The ARTWare Components

DOMAIN-ORIENTED MULTIMEDIA AUTHORING

In order to support users to perform their tasks using multimedia objects, we need domain-oriented interactive application systems [6]. Although a number of multimedia application systems have been available as commercial packages, shareware, and freeware, those systems do not often best serve for their tasks.

Let us take an empirical video analysis task, for instance. Early stages of empirical video analysis task that usability testers and requirements analysts conduct are best supported by viewing it as an exploratory task rather than by applying standard statistical analysis techniques [10].

Most of existing tools that help users understand video data are automated video summarization tools that help users more quickly browse video clips [2] [4] [14], or video annotations tools that aim at supporting a student to learn about the content of the videos [5][9]. None of existing systems support a specific task of empirical video analysis in the manner stated above.

In order to overcome such situations, therefore, domain specialists need to build or ask someone to build domain-oriented multimedia authoring environments that support their tasks.

Building such a system, however, is very "expensive." It requires a wide range of development knowledge. In addition to regular programming skills, the developers need to have knowledge about topology and geometry to deal with 3D graphics, about colors and fonts and their cognitive effects, about low-level implementation of movies and sound that are often hardware-dependent, and about how to design user interface based on theories in human-computer interaction.

OUR APPROACH

Our approach to help such situation is to provide a library of components for building domain-oriented interactive multimedia authoring environments. The essential part of our approach is two fold: design and integration.

Appeared in the Proceedings of International Conference on Future Software Technology (ISFST) 2001, Software Engineers Associates, ZhengZhou, China, pp. 246-251, November, 2001.

Components Design

Components for such a library must be well designed based on theories of human-computer interaction. We argue that externalizations play a critical role in intellectual creative tasks [3]. Properly designed representations make the same task much simpler than other representations [11].

The components of ARTWare are designed based on the concept called ART (Amplifying Representational Talkback) [16]. The concept emphasizes the importance of externalization in supporting cognition-intensive tasks [18], and ART uses perceptual feedback to support users in externalizing problem situations [1].

Specifically, ARTWare provides components that use spatial positioning as a way to collect and annotate objects that a user thinks important or interesting while viewing them [13]. Using the space, a user may put two objects closer to each other to represent they are related to each other, may put one object on top of another to represent the former is more important than the other, or may make an object much larger than others to represent that the object is by far more important than others.

Components Integration

Many of the components of ARTWare can be easily integrated by synchronizing behavior of objects across different components. ARTWare is built on VisualWorks Smalltalk, a pure object-oriented programming language. ARTWare adheres to a pure object-oriented MVC (Model-View-Controller) architecture [8], therefore it is straightforward for the system builder to have a single object displayed in multiple presentation styles in a synchronized manner.

In many of such tasks that use multimedia objects, users need to examine objects from a variety of perspectives. For empirical video analysis tasks, for instance, it is important that an analyst can easily view multiple movies in a synchronized manner.

However, integrating components often require low-level programming. Once a component is built without much consideration of integration with other components, it is often not easy to add interfaces for integration later and consequently, it becomes necessary to rebuild the component from scratch.

To avoid this situation, our components have been built based on the carefully manifested architecture. Each of ARTWare components has been built with an assumption that they will be integrated with others. As a consequence, each of the components can be put together into a single system with very little effort.

ARTWARE COMPONENT LIBRARY

This section gives detailed description of major components of ARTWare. Although each component can be used as individual tools, the power of ARTWare resides where data and controls are shared among multiple components. Figure 1 illustrates major components of ARTWare.

MultimediaViewer

A MultimediaViewer can display any of a text file, a jpeg image file, a QuickTime sound file and a QuickTime movie file. Movie and sound can be cropped (segmented) by using the two bars located at the bottom.

In designing MultimediaViewers, we have decided to provide user interfaces that help a user: (1) crop parts of a movie (sound) file as easy as possible, (2) know where each of cropped parts comes from, and (3) re-edit a cropped part by shrinking or extending.

The two horizontal controller bars, the p (proportional)-bar and f (fixed)-bar helps a user know where the cropped part comes from in terms of the original movie (sound), and shrink or extend the cropped part in fine detail. The cropped part is indicated with a green line segment in terms of the whole movie on p-bar.

MovieTracker

In tasks such as video analysis, it is often necessary to simultaneously play multiple movies to compare the contents. MovieTracker allows a user to play multiple movies in a synchronized manner.

MovieTrailer

In editing a movie or analyzing the contents of a movie in fine detail, a user often needs to examine a movie very carefully by knowing what has just happened and what is

going to happen next. MovieTrailer is a movie viewer, consisting of five movie screens. When playing, the five screens from the left plays the movie in succession with some delay time.

MovieBelt and MovieEddy

In addition to movie viewers, ARTWare provides two types of movie frame-sequence viewers: MovieBelt and MovieEddy.

When a user starts a MovieBelt or a MovieEddy from one of MultimediaViewer or MovieTrailer, the system asks the user how many frames that the user would like to have. When specified, the system creates the specified number of frames out of the original movie, and creates a sequence of the movie frames. MovieBelt shows the frames arranged in a straight line and MovieEddy shows the frames in a spiral shape depicted in a 3D space.

ElementSpace, Viewfinder and TimeChart

As described above, ARTWare components are designed based on the ART concept and uses spatial positioning as a way of externalization. ElementSpace is a space where a user can position a cropped movie, sound, image or edited text produced by MultimediaViewers, MovieTracker and MovieTrailer described above.

Two design concerns have been addressed.

First, each object positioned in the space needs to be identified as easily as possible. The ART concept emphasizes the use of perceptual information to help this process [7].

Second, ElementSpace is implemented as a 3D space, not as a 2D space, and emulates a 2D space. The design rationale to make this decision is a matter of implementation; the computational speed. In designing ARTWare, it is critical that a user can move and resize objects as natural as possible. Manipulation of objects placed on the 2D implementation is done by bitmap operations, which is very expensive. In contrast, once objects are put in a 3D space, manipulation of the objects is done by translation, revolution and texture mapping. Although the 3D implementation requires a much larger memory resource, its computational speed outweighs that of the 2D implementation.

The two 3D views, Viewfinder and TimeChart, both dynamically display objects at the X-Y coordinates in ElementSpace and use the Z-axis differently. The control over the objects in ElementSpace is synchronized with the two 3D views.

DocumentViewer

Not only visual annotation by positioning, but also textual annotation is also supported in ARTWare. A user may associate a text annotation with any type of the displayed objects in ElementSpace. When a user selects an object in the ElementSpace, and selects the Annotate command from the menu, a simple text editor is opened and a user can type in text as an annotation for the object. These annotations are not displayed in ElementSpace, but are displayed in DocumentViewer.

DocumentViewer provides a synthesized view of objects positioned in the ElementSpace. DocumentViewer displays objects in ElementSpace in a sequential order but in a table format. The left column displays objects from top to bottom in the order specified in terms of the three axes, and the right column shows each annotation associated with the object displayed to the left.

INTERACTIVE TOOLS USING ARTWARE COMPONENTS

We have instantiated interactive systems for multimedia objects by integrating ARTWare components in different types of tasks.

ART#001: A Tool for Text Editing

The ART#001 system is a text editor for early stages of writing [15][16]. The system does not use any multimedia objects other than text. Using ART#001, a user gradually constructs a document by editing text and creating a text segment, positioning text segments in ElementSpace while rearranging them, and gradually constructing a document.

Figure 2 shows a screen image of ART#001. A user can edit a text using the window at the bottom and put the segment in the space on the top right window. Text segments in the space can be copied, merged, split, or deleted. The top-left window, DocumentViewer, appends all the text segments in

Appeared in the Proceedings of International Conference on Future Software Technology (ISFST) 2001, Software Engineers Associates, ZhengZhou, China, pp. 246-251, November, 2001.

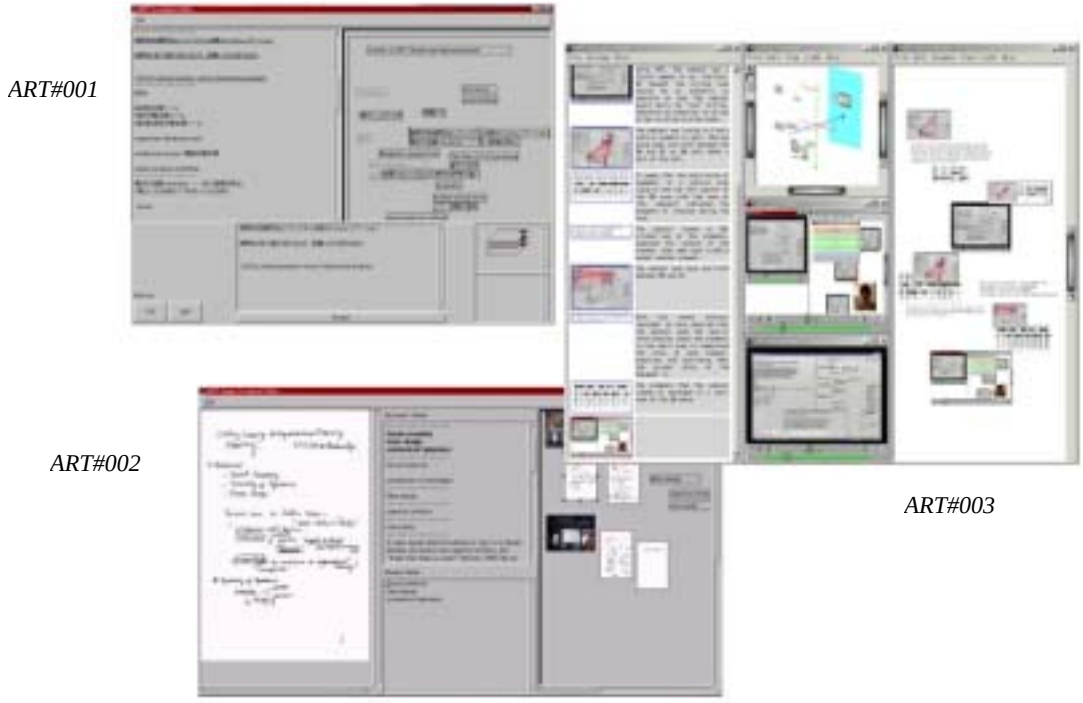

Figure 2: The ARTWare Systems

the space from top to bottom.

ART#002: A Tool for Note Annotating

There is often a case that we need to develop a document out of notes taken during meetings for a particular purpose (for instance, identifying software requirements specifications). The need to integrate the use of hand-written notes into a document writing process has emerged naturally based on this context.

ART#002 system (Figure 2) helps a user synthesizing hand-written notes by positioning the notes in the ElementSpace (as thumb-nail images) and gradually adding textual descriptions and interpretations to the notes. The system supports a process of constructing a well-formulated document out of hand-written note images for the purpose of less-ambiguous, clearer communication among the document readers such as members of a software project.

ART#003: A Tool for Video Data Analysis

ART#003 is an interactive system that supports empirical video-data analysis [17]. The system helps software engineering experimenters analyze video-taped user data and other types of empirical data (such as keystrokes, and eye movements).

ART#003 is developed based on ARTWare to help experimenters discover important aspects of the data, to collect them, to store them and to share them with peers. As shown in Figure 2, a user of ART#003 can browse synchronized multiple video data, crop seemingly interesting parts and put them in ElementSpace..

DISCUSSIONS

There have been powerful and high-functional systems for manipulating multimedia objects on computer systems. For instance, much research has been done on video summarization [2][9] and spatial positioning [12]. We do not argue that ARTWare outweighs such existing systems in particular domains. Rather, our goal has been to develop a library of components that are general enough so that users and developers can adapt the components for their specific purposes.

By reporting our effort here, we do not mean that ARTWare provides a complete set of necessary components. To the contrary, we think it is important that the library will be

evolved by having users instantiate ARTWare and receiving requests from them.

CONCLUSION

ARTWare has been developed as a middleware for the development of domain-oriented authoring environment. Each component is carefully designed based on the theory of human-computer interaction, and the control and data among the components are shared so that it is easy to obtain the feeling of integration by using the components. ARTWare consists of interactive components which users can either use as individual tools, or combine them to construct a domain-oriented authoring system that serves for their specific needs.

ACKNOWLEDGMENTS

This study was supported by the Proposal-based New Industry Creative Type Technology R&D Promotion Program from the New Energy and Industrial Technology Development Organization (NEDO) of Japan.

REFERENCES

1. R. Arnheim, Visual Thinking, University of California Press, CA, 1969.
2. J. Boreczky, A. Girgensohn, G. Golovchinsky, S. Uchihashi, An Interactive Comic Book Presentation for Exploring Video, Proc. of CHI2000, ACM Press, pp.185-192, 2000.
3. J. Bruner, The Culture of Education (Harvard University Press, Cambridge, MA, 1996).
4. A. Cheyer, L. Julia, MVIEWS: Multimodal Tools for the Video Analyst, Proceedings of IUI 98, pp.55-62, 1998.
5. A. Dekel, D.O. Bergman, Synopsus: A Personal Summary Tool for Video, Extended Abstracts of CHI2000, ACM Press, pp.4-5, 2000.
6. G. Fischer, Domain-Oriented Design Environments, Automated Software Engineering, Kluwer Academic Publishers, Vol.1, Boston, MA., pp.177-203, 1994.
7. B. Indurkhya, On Creation of Features and Change of Representation, Cognitive Studies, Vol.5, No.2, pp.43-56, June 1998.
8. S. Lewis, The Art and Science of Smalltalk, Prentice Hall, 1995.
9. F. C. Li, A. Gupta, E. Sanocki, L. W. He, Y. Rui, Browsing Digital Video, Proc. of CHI2000, ACM Press, pp.169-176, 2000.
10. W. E. Mackay, M. Beaudouin-Lafon, DIVA: Exploratory Data Analysis with Multimedia Streams, Proc. of CHI'98, ACM Press, pp.416-423.
11. D. Norman, Things That Make Us Smart (Addison-Wesley Pub. Co., MA 1993).
12. G. Robertson, M. Czerwinski, K. Larson, D. Robbins, D. Thiel, M. van Dantzich, Data Mountain: Using Spatial Memory for Document Management, Proceedings of UIST '98, 11th Annual Symposium on User Interface Software and Technology, ACM, New York, NY, pp. 153-162, 1998.
13. S. Takada, Y. Yamamoto, K. Nakakoji, Two-Dimensional Positioning as Visual Thinking, Theory and Application of Diagrams (Diagrams2000, Edinburgh, UK), M. Anderson, P. Cheng, V. Haarslev (Eds.), Springer-Verlag, Berlin, pp.437-452, September, 2000.
14. S. Uchihashi, J. Foote, A. Girgensohn, J. Boreczky, Video Manga: Generating Semantically Meaningful Video Summaries, Proceedings of ACM Multimedia 99, ACM Press, 1999, pp.383-392.
15. Y. Yamamoto, S. Takada, M.D. Gross, K. Nakakoji, Representational Talkback: An Approach to Support Writing as Design, Proceedings of the APCHI '98 Conference (Kanagawa, Japan), IEEE Press, pp. 125-131, July, 1998.
16. Y. Yamamoto, K. Nakakoji, S. Takada, Hands-on Representations in a Two-Dimensional Space for Early Stages of Design, Knowledge-Based Systems Journal, Elsevier Science, Vol.13, No.6, pp.375-384, November, 2000.
17. Y. Yamamoto, A. Aoki, K. Nakakoji, Time-ART: A Tool for Segmenting and Annotating Multimedia Data in Early Stages of Exploratory Analysis, CHI2001, Extended Abstract, April 2001 (in print).
18. J. Zhang, The Nature of External Representations in Problem Solving, in: Cognitive Science, Vol. 21, No.2, pp. 179-217, 1997.