Enriching Hypermedia Application Interfaces

André T. S. Fialho and Daniel Schwabe

Departamento de Informática – Pontifícia Universidade Católica do Rio de Janeiro
(PUC-Rio) – Caixa Postal 38.097 – 22.453-900 – Rio de Janeiro – RJ – Brazil
atfialho@inf.puc-rio.br, dschwabe@inf.puc-rio.br

Abstract. This paper presents a systematic approach for the authoring of animated multimedia transitions in Web applications, following the current trend of rich interfaces. The transitions are defined based on an abstract interface specification, over which a rhetorical structure is overlaid. This structure is then rendered over concrete interfaces by applying rhetorical style sheets, which define concrete animation schemes. The resulting applications has different transition animations defined according the type of navigation being carried out, always emphasizing the semantically important information. Preliminary evaluation indicates better user experience in using these interfaces.

1 Introduction

Current web applications have become increasingly more complex, and their interfaces correspondingly more sophisticated. A noticeable tendency is the introduction of animation as an integral part of Web application interfaces, after the advent of AJAX technologies (see, for instance, the Yahoo Design Patterns Library for transitions [7]). In hypermedia applications a more complex kind of animation is involved when considering entire interface changes that occur during navigation, where there is a transition as a result of a navigational operation. As such, this type of interface change is a prime candidate for the application of animation techniques.

This paper presents an approach for systematically enriching hypermedia applications by extending the SHDM approach [3]. In particular, attention is paid on how to relate animations to the application semantics expressed in the SHDM models.

The remainder of this paper is organized as follows. Section 2 gives some background on the use of animation, and on the representation of interfaces in SHDM. Section 3 presents the proposed approach, and Section 4 presents a discussion about the results and conclusions.

2 Background

The common definition of animation is the result of several static images that, when exhibited in sequence, creates an illusion of continuity and movement. Nowadays there are several systems that use animation with the purpose of enriching the interaction process and the ser experience or even provide smooth transitions the prime examples being the MacOS, and more recently Windows Vista. Some experiments

L. Baresi, P. Fraternali, and G.-J. Houben (Eds.): ICWE 2007, LNCS 4607, pp. 188 – 193, 2007.

indicate that animations can be of help [1]. Furthermore, film editing knowledge has been applied to the design of human-computer interfaces [5].

Since animations will be expressed in terms of interface elements, we first summarize how the interface is specified in SHDM through Abstract and Concrete Interface Models [6]. The abstract interface model is built by defining the perceptible interface widgets. Interface widgets are defined as aggregations of primitive widget (such as text fields and buttons) and recursively of interface widgets. Navigational objects are mapped onto abstract interface widgets. This mapping gives them a perceptive appearance and also defines which objects will activate navigation.

The Abstract Interface is specified using the Abstract Widget Ontology, which establishes the vocabulary. According to it, an abstract interface widget is can be a Simple Activator, which is capable of reacting to external events, such as mouse clicks; an Element Exhibitor, which is able to exhibit some type of content, such as text or images; or a Capturer, which allows input of data, including widgets like input text fields, and selection widgets such as pull-down menus and checkboxes, etc... Finally, it can also be a CompositeInterfaceElement, which is a composition of any of the above.

Once the Abstract Interface has been defined, each element must be mapped onto both a navigation element, which will provide its contents, and a concrete interface widget, which will actually implement it in a given runtime environment.

3 Introducing Animations in Hypermedia Applications

The process for the insertion of animations during the design of the application is composed of four stages illustrated in Fig. 1, in which each of the stages produces a specific output used by the next stage.

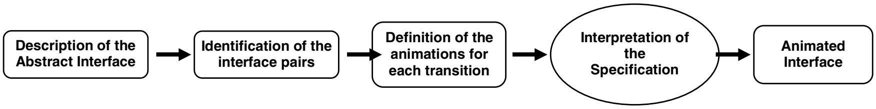
Description of the Abstract Interface
Identification of the interface pairs
Definition of the animations for each transition
Interpretation of the Specification
Animated Interface

Fig. 1. Steps to produce an animated interface

The animations proposed in this work are displayed to the user during interactions that define a change in the navigational state. Each change is relative to a pair of distinct source/destination interfaces that represent these corresponding navigation states. We call this process a transition, which can be represented as an intermediate animated interface between two interfaces. This intermediate interface is only representational, and is described as a list of animations.

Each interface is a composition of widgets. A transition animation between interfaces is the process of visual transformation that transforms the source interface into the destination interface. Therefore, the transformation to be applied to each widget has to be specified.

To set up an interface animation it is necessary to identify the abstract widgets that compose each interface, and then specify which pairs of interfaces will define the source and destination of the transition. This is determined by the navigational structure of the application and associated abstract interfaces, as specified in the SHDM model of the application.

For each defined transition we need to identify widgets (in the source and destination interfaces) that are mapped to the same or related element in the model. As a result, we identify which widgets remain unchanged, which disappear, which appear, and define which widgets are related. The first three behaviors are straightforward; the last one will depend on which relationship the designer whishes to expose. A common example of this last widget relation is when widgets in the source and destination interfaces are mapped to different attributes of the same element in the model (e.g., a name and a picture).

After pairing the widgets, we must provide the transition specification for the navigational change. The transition specification is made considering a pre-defined set of animation functions, identified considering that only three basic actions can be applied to a widget: a removal, an insertion or a transformation, which can be a

- Match – For widgets that are identified as remaining in the destination interface (i.e. they are present in both the source and the destination interfaces), it is necessary to match their appearance parameters such as position, size and color. This transformation animation responsible for matching these parameters.
- Trade – A transformation animation responsible for exposing the relation between two distinct related widgets during the transition, for example as a morph.
- Replace – When the same widget exists in the source and destination interfaces, but the associated elements in the model are different, this transformation animation replaces the information contained within a widget.
- Emphasize – A transformation animation that alters certain parameters such as size or color of a widget to emphasize an element.

Each of these functions will also have properties that describe the point in time it should occur within the transition, and which effect should be used (fade, push, grow, etc). These properties are specified according their role in the transition, described next.

3.1 Rhetorical Animation Structure

When we define the transition specification we must describe not only the list of the animation actions that will occur, but in which order in the timeline they will be executed, animation effect and the duration of each action. This sequence in which the animations are presented has great importance since it influences how the transition will be interpreted by the user.

In order to determine the best sequence and which effects should be used in each animation we propose the use of a rhetorical animation structure. This approach is inspired by the use of Rhetorical Structure Theory (RST) [4], as it has been used for generating animation plans ([2]). With this structure we can define the communicative role of each animation during the transition, and so identify which animations are more important and how they should be presented to better inform the user of the transformations that occur.

The rhetorical structure is specified in terms of rhetorical categories, which classify the various possible animation structures, as follows:

- Removal – Set of all animations that achieve an element removal (widgets that disappears). Rhetorically, these animations clean up the screen to set the stage for the upcoming transition;
- Widget Feedback – Any kind of transformation that represents an immediate feedback of the triggered widget. Rhetorically, these animations emphasize that the request made has been received, and the application is about to act on it.
- Layout animations – Set of animations that change (insert or transform) interface widgets that are independent of the contents being exhibited. These widgets are typically labels, borders, background images, etc…
- Secondary animations – Set of animations that transform interface widgets associated to secondary (satellite in terms of RST) elements
- Main animations – Set of animations that transform interface widgets associated to main (nucleus in terms of RST) elements.

Once the structures are chosen the designer must categorize the animation functions that have been identified in the previous step. We can partially aid this classification by observing the navigational model, identifying which relations are more important to describe. For example, transitions between objects of different classes should help identify the relation and the contexts associated with the navigation step being carried out in the transition.

The next step after the functions have been allocated to the rhetorical categories is to determine a rhetorical structure in which the animations will be presented. Different sequences can be arranged for each type of navigation. For example, Fig. 2 shows one possible sequence using these rhetorical categories.

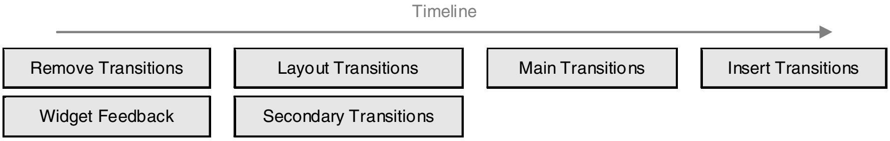

Fig. 2. Rhetorical animation structure

This sequence follows the rationale that first the screen should be cleared of elements that will disappear, simultaneously with a feedback of the activated widget. Next, the screen layout is changed to reflect the destination interface, in parallel with the secondary transitions (i.e., those that are judged as accessory to the main transition) are made. Then the main transition is carried out, as the most important part, followed by the insertion of new element.

After defining the rhetorical animation structure we need to map the categories into concrete transitions that describe which are the effects, duration and the sequence of the actions within the structure. The specification is done through a set of styles defined as a Rhetorical Style Sheet which reflects the designer preferences, and can be guided by the use of specific patterns that gather solutions to common transition problems within a specific context.

3.2 Implementation

The next step once the specification is done is to interpret this specification so the animations are presented to the user during the interaction. In this work we use an environment for supporting animation on web documents, in which HTML web pages represent the different types of interfaces, and JavaScript technology using dynamic HTML for the animations. Fig. 3 shows a diagram with the sequence of events in the implemented environment.

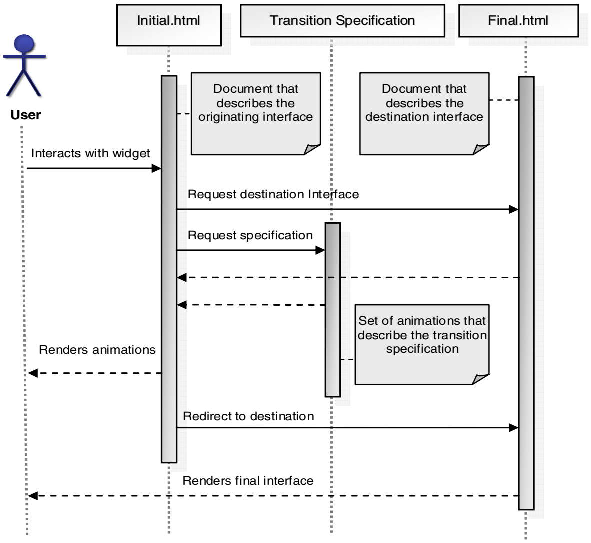
Initial.html
Transition Specification
Final.html
User
Document that describes the originating interface
Document that describes the destination interface
Interacts with widget
Request destination Interface
Request specification
Set of animations that describe the transition specification
Renders animations
Redirect to destination
Renders final interface

Fig. 3. Diagram representing the environment

Given the reduced space available, and the nature of this work, we have developed a demo flash application of an hypermedia movie database application using the approach described in this document, with step-through functionality to help understand the rhetorical animation structure being followed.. This example can be accessed at http://www.inf.puc-rio.br/~atfialho/hmdb/hmdb.html (requires a flash plug-in to execute).

4 Conclusions

This paper presented an approach for adding animation to hypermedia applications, enriching a set of existing models in SHDM. Although several initiatives exist to add animation to web pages, we are not aware of any published description of approaches dealing with entire web page transitions.

We have so far made only informal evaluations of the resulting interfaces obtained through this approach. Users have given positive feedback about the so-called “user experience”, and seem to prefer animated interfaces over equivalent non-animated interfaces. However, a more systematic evaluation will still be carried out.

While based on models and being more structured, the present approach still poses authoring difficulties, since they require manual insertion and choice of animation effects for each interface widget. We are currently investigating the use of wizards and the construction of a Rhetorical Style Sheet library to aid designers for the more common tasks routinely encountered in designing hypermedia applications.

Acknowledgement. Daniel Schwabe was partially supported by a grant from CNPq.

References

1. Bederson, B.B., Boltman, A.: Does Animation Help Users Build Mental Maps of Spatial Information? In: InfoVis '99. Proceedings of IEEE Symposium on Information Visualization '99, pp. 28–35. IEEE Computer Society Press, Los Alamitos (1999)
2. Kennedy, K., Mercer, R.E.: Using Communicative Acts to Plan the Cinematographic Structure of Animations. In: Cohen, R., Spencer, B. (eds.) LNCS (LNAI), vol. 2338, pp. 132–146. Springer, Heidelberg (2002)
3. Lima, F., Schwabe, D.: Application Modeling for the Semantic Web. In: Proceedings of LA-Web, Santiago, Chile, Nov. 2003, pp. 93–102, IEEE Press,ISBN (2003), available at http://www.la-web.org
4. Mann, W.S., Thompson, S.: Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. Text, 8(13), 243–281 (1988)
5. May, J., Dean, M.P., e Barnard, P.J.: Using Film Cutting Techniques in Interface Design. In: Human-Computer Interaction, vol. 18, pp. 325–372. Lawrence Erlbaum Associates, Inc., Mahwah, NJ (2003)
6. Moura, S.S., Schwabe, D.: Interface Development for Hypermedia Applications in the Semantic Web. Proc. of LA Web, Ribeirão Preto, Brasil, pp. 106–113. IEEE CS Press, Los Alamito (2004)
7. Yahoo Design Patterns Library (transitions), available at http://developer.yahoo.com/ypatterns/parent.php?pattern=transition