

TagVisor: Extending Web Pages with Interaction Events to Support Presentation in Digital Signage

Marcio dos Santos Galli
TelaSocial
São Carlos, Brazil
mgalli@telasocial.com

Eduardo Pezutti Beletato dos Santos
Araraquara Center University-UNIARA
Araraquara, Brazil
eduardo@epbsantos.com

ABSTRACT

New interaction experiences are fundamentally changing the way we interact with the web. Emerging touch-based devices and a variety of web-connected appliances represents challenges that prevents the seamless reach of web resources originally tailored for the standard browser experience. This paper explores how web pages can be re-purposed and become interactive presentations that effectively supports communication in scenarios such as digital signage and other presentation use cases. We will cover the TagVisor project which is a JavaScript run-time that uses modern animation effects and provides an HTML5 extension approach to support the authoring of visual narratives using plain web pages.

Categories and Subject Descriptors

H.5 [Information Interfaces and Presentation]:
H.5.2 User Interfaces
H.5.4 Hypertext/Hypermedia User Issues

General Terms

Human factors, algorithms, design

Keywords

W3C; DOM; HTML5; CSS3; JavaScript; Digital Signage; Web; Boot2Gecko

1. INTRODUCTION

We present several methods that conveys the use of web standards and proposes that web pages can be enriched with meta-data that enables rich animation experiences, thus extending the use of web content into new interaction experiences beyond the standard desktop browsing. In this context, the TagVisor [9] project showcases experimental scenarios and a practical implementation of a run-time system that can be attached to existing web content, thus enabling visual and interactive narratives that can be used in digital signage or live presentations.

The first scenario is the guided tour experience and it implies the use of transformation effects and time-referenced

data in order to build a visual narrative out of regions of a web page. Our first demonstration refers to an authoring experience where the user can mark elements of interest within the page - namely anchors. These anchors are arranged in a time-line to form a visual narrative using transitions effects. The main effects implemented in TagVisor are zoom and pan which are transformations that can be accomplished using CSS 3 transforms [1]. This paper covers details about these effects because the combination of both can create the idea of a camera, which supports the guided tour metaphor.

The guided tour scenario is presented as model for authoring and adaptation of web content that can be used in digital signage screens, thus becoming a cheaper, lightweight, and elegant solution for communication in open spaces.

In another scenario, we will explore how the guided tour case can be extended with limited interactivity. Our study case is going to be the act of presenting a topic performed by a mediator - a professor. This model follows from the guided tour but it is extended with interaction controls so the presentation can be executed with intended or unintended interruptions which are common in the classroom environment. In this context, we will examine the metaphor of slideshow systems, such as Prezi¹ and propose a simple model that adapts the guided tour case to be able to serve in these scenarios with interactivity.

In the final section, we will conclude bringing concepts such as virtual videography [11] and a direction for future research work with focus in kinds of visual effects and its uses regarding the kinds of documents - an aim to support the creation of visual and interactive narratives that can be automatically generated using external data-sources such as a reference base of known presentations.

2. RELATED WORKS

The zooming web browser project [7] proposes the use of dynamic multi scale techniques to support users with focus and context when navigating. On the other hand, it's known that context and interaction techniques [13] can be necessary when users are lost using zoomable interfaces for navigation. These problems regarding context and interaction are more evident in cases where direct navigation is needed.

As interaction is not considered in the guided tour case, we highlight the use of CSS3 transformations [1] which is important to produce the pan and zoom effect in the browser engine significantly faster than using JavaScript. However,

¹Prezi uses zoom and animation metaphor for presentations with Flash technology - <http://www.prezi.com>

TagVisor Video Demonstration:
<http://labs.telasocial.com/tagvisor-demo>

Copyright is held by the International World Wide Web Conference Committee (IW3C2). IW3C2 reserves the right to provide a hyperlink to the author's site if the Material is used in electronic media.
WWW 2013 Companion, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2038-2/13/05.

events of transformation, such as insertion of style, relies in DOM 1 [3] interfaces using JavaScript which ensures time-based control.

For the extended scenario, as we evolve into the live presentation case, context and interaction techniques are introduced and known cases of modern presentation systems should be presented. These systems are known from long date and some of the early web-based solutions were available as templates for web developers. One of first was a slideshow presentation system known as **S5: A Simple Standards-Based Slide Show System** [12] created by Eric A. Meyer. S5 is entirely based in XHTML, CSS, and JavaScript and can run a complete slide show and have a printer-friendly version.

A modern example is **html5slides** [4], maintained by Google and developers using HTML5 and CSS3. Slides are defined in article tags and it's possible to add extra stylesheets to override the default CSS. Within the HTML5-compatible solutions, we highlight **Reveal.js** [10] created by Hakim El Hattab, which provides a great look and can be positioned horizontally or vertically. It's also easy to change the CSS to add your own effects and slides are defined in separate section tags and Reveal.js offers a choice of several 3D slide transitions. Yet another known solution is a library named **Impress.js** [14] created by Bartek Szopka. It's been inspired by the commercial Prezi slides, however it produces a three dimensional space so slides will pan, scale and rotate using a coordinate system in three dimensions.

3. SCENARIO 1: GUIDED TOUR WITH HTML5

Our initial focus is to positively impact the production of contextual animations, namely visual narratives, therefore supporting a viable model for creation of web-based content to mediums such as display appliances in physical spaces.

We refer to the theoretical basis [8] for multi scale interfaces and propose a standards compliant model to support zooming for specific sections in an HTML document. Figure 1 shows a web page and three possible views for the same user. View A refers to a visualisation for a section at the top of the document - say the focus is in the title. View B points to another section, for example the paragraph under that title. If a smooth animation is proposed from A and B, the expected effect will involve panning because the distance from viewports A or B and the document plane are the same. The C view is a different case and the animation from viewport B to C implies in a zoom out effect mainly because A is at a higher distance from the observable plane.

3.1 CSS3 Transforms for Pan and Zoom

The combined transformation model for pan and zoom is accomplished using CSS 3 transforms for translate and scale applied respectively to the BODY and HTML elements. You can think of the visible browser window as being the viewport - the observer. First the document's HTML element receives a style attribute **overflow:hidden**, thus forcing scrollbars to go away and the HTML element becomes a rectangle and viewport to the inner body element.

After that, two calculations will take place based in the target interest area which is the target element in the figure 2. For the two transformations, B1 and H1, the dimensions

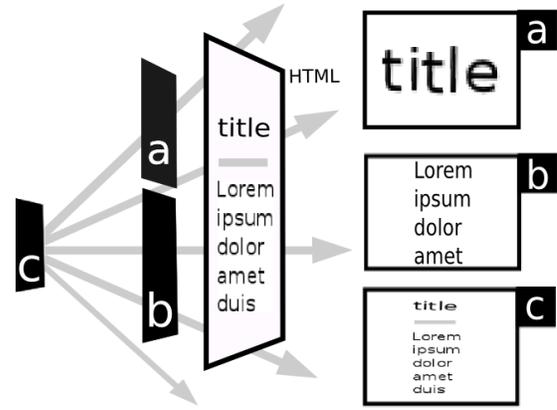


Figure 1: A, B, and C are possible viewports. A to B can be achieved with panning but C involves pan and zoom out

for the target element and the viewport (window) are fundamental to achieve the pan and zoom effects.

For B1, a CSS3 translate operation is applied to the document's body. The following expressions are used to obtain left and top coordinates for the translate transformation:

$$X_b = - \left(X_t - \frac{W_v - W_t}{2} \right)$$

and

$$Y_b = - \left(Y_t - \frac{H_v - H_t}{2} \right),$$

where X_t, Y_t, W_t, H_t refers to the target's left, top, width, and height. And W_v and H_v refers to the viewport's width and height.

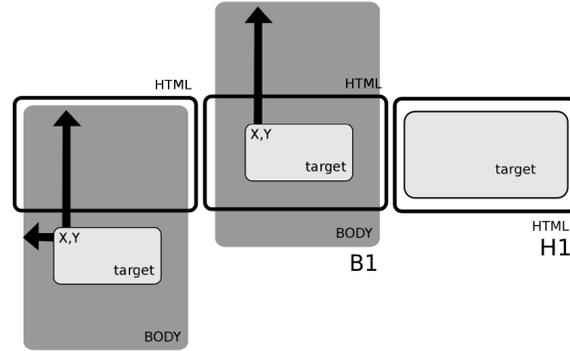


Figure 2: B1 represents a translate operation in the body until the target element reaches the center; H1 applies a scale in the HTML element until the target element fits the window

For H1, a CSS3 transformation, namely scale, will be applied to the HTML element to scale the viewport so that the target element fits within. The scale factors, S_W and S_H , are given by the following expressions:

$$S_W = \frac{W_v}{W_t} \text{ and } S_H = \frac{H_v}{H_t}.$$

We need to apply one of the scale factors to ensure that the aspect ratio is preserved. We pick S_W or S_H after testing which of the scale transforms will ensure that the target's rectangle fits the viewport.

3.2 HTML5 Custom Data for Animations

The state diagram in figure 3 illustrates a guided tour example for a viewport animation that zooms among three DIV elements in a page. Each **zoom-to** arrow represents the pan and zoom effect that is executed as a function of time.

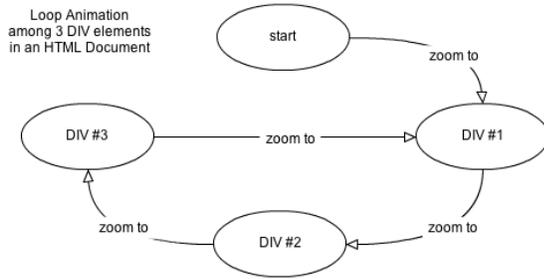


Figure 3: A state diagram for a sequential loop animation among DIV elements

In order to store an animation, we use HTML5 custom data using **data-*** attributes. These attributes are non-visible per specification [2] and placed in **li** elements under a hidden **ul** container element.

```
<ul id="animation" style="display: none;">
  <li data-target="slidea" data-time="1s"
      data-effect="scalefit" data-duration="3s">
  </li>
  <li data-target="slideb" data-time="6s"
      data-effect="scalefit" data-duration="3s">
  </li>
  <li data-target="slidec" data-time="12s"
      data-effect="scalefit" data-duration="3s">
  </li>
</ul>
```

3.3 Web Animation Run-time in Digital Signage

With a proposed web standards infra-structure in place, we identified a viable model for content distribution and communication that reuses web assets and suggests a light-weight approach for data distribution and yet can ensure a reasonable visual experience in modern digital signage. The TagVisor animation run-time is being tested as part of a standards-compliant digital signage solution named TelaSocial [6]. TelaSocial is a powered by Mozilla [5] application that is launched directly at the startup process in the operating system. This high level model is also found in Boot2Gecko² by Mozilla. Figure 4 is an overview of the web engine infrastructure that is launched after the boot process in a Linux computer: TelaSocial is highlighted at the top as

²Book2Gecko is Mozilla's modern technology that launches a web engine and additional application processes directly in boot time, thus exposing web functionality in modern appliances.

an application that brings a browser function in full-screen. The initial start point is local or a remote HTML page. In the simplest test scenario, we can run a web page that has a TagVisor-compatible script which initiates the animated experience.

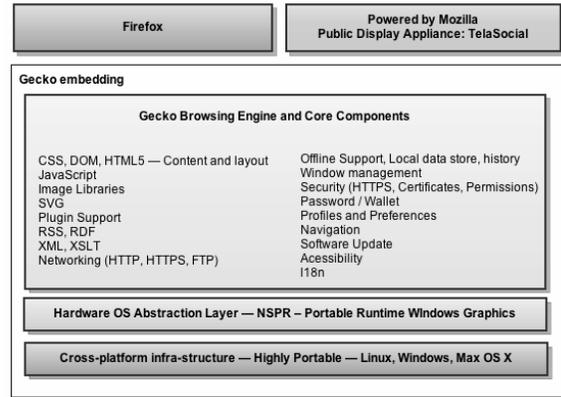


Figure 4: Overview of Gecko infra-structure and kiosk application which brings Web content

4. SCENARIO 2: EXTENDED INTERACTION

Consider a scenario when a professor gives a lecture using a deck of slides. While the order arrangement may serve as a guideline to the narrative, it is also expected to happen interactions, such as interruptions prompted by students or a planned interruption when the professor asks the audience. In this case, the possibilities for interaction are known just like the guided tour requires a known script for its animation.

Since the guided tour resource is not made of fixed frames, such as the case with static videos, such markup driven and dynamic model represents the basis for a formal model that supports interactivity while incorporating the the kind of events supported by the guided tour. Figure 5 shows an example of this hybrid model:

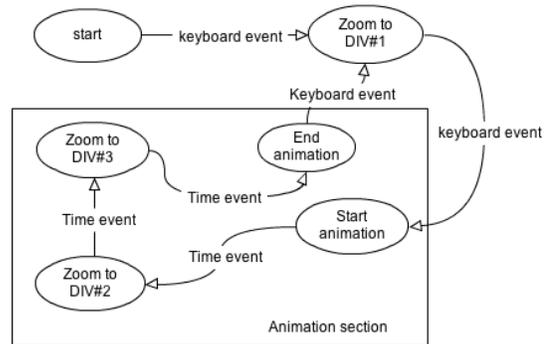


Figure 5: Diagram with keyboard event interruptions between occurrences of the guided tour animation

The above case is a simplified example where interactions are known. Consider an example of a presentation that runs

continuously until the moment an user touches the screen. When the user touches the screen, the automatic execution stops and the application runtime is turned into a slide show, thus listening to user's touch events for next or previous slides. In addition, other interactions are expected, such as a means for the user to see all the slides in the visible screen, thus allowing the user to touch over a slide as a shortcut to reach specific presentation sections. With this complex case, we bring the notion of contextual states that can modify how inner states are executed in a given graph diagram. This general notion implies that the kind of arrows, i.e. the events between states, can be switched depending on the context. Such model yields to interesting opportunities. As an example, an author could debug a presentation - for example via controlling a step by step execution instead being locked with a time-referenced execution. It's also possible to re-run the animation at different speeds or switch an input method context while maintaining the script.

In another demonstration scenario we show an interactive slides player where keyboard events are switched by web services-driven messages, thus enabling a mobile application to be used as a remote control to control a guided tour interactively in the screen.

5. CONCLUSION AND FUTURE WORK

In this work, we initially highlight the impact towards improving context and focus for content that is targeted to end-users using digital signage systems. However, our proposed model can be used to allow users to modify web content and build interactive narratives yet preserving the markup.

With an initial focus in the observer, i.e. the notion of a viewpoint with zooming, we believe that other videography elements can be used to improve the authoring experience in visual narratives. Virtual videography [11] demonstrates that a decision making framework and system techniques can be used to automatically record a professor's lecture in video; and that rules and elements of videography can impact the effectiveness of recorded lectures for education. At this stage, we also combined other visual effects currently integrated as part of the TagVisor library itself. Some of these elements are: fade in and out, scrolling or typing text widget, 3D rotation, and more.

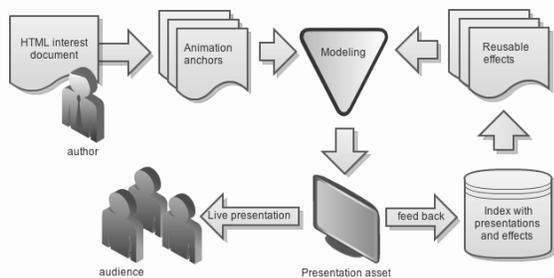


Figure 6: Visual narrative meta-data builds an index system and facilitates the reuse presentation assets

Figure 6 presents an overview for future works. We aim to evaluate what kinds of elements of videography and techniques are more effective regarding the kinds of pages. Since

each narrative has meta-data associated with these documents, it is possible to feed a knowledge base, thus to help users in reuse visual effects and support the decision making process as they can access, modify and evolve from examples. The dynamic nature of this model should be an opportunity that fosters the development of new narratives out of existing pages and linked narratives.

Acknowledgments

We appreciate the support from Mauricio Araldi for contribution to keyboard events and a focus to presentation. Juliana Cobre for the math review. And for Itaipu Technological Park in Brazil, for their investments in the TagVisor/TelaSocial project and works with interaction³ through the open-source project.

6. REFERENCES

- [1] Css3 transforms. <http://www.w3.org/TR/css3-transforms>.
- [2] Custom data attributes. <http://www.w3.org/TR/2011/WD-htm15-20110525/elements.html>.
- [3] Document object model (dom) level 1 specification. <http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001>.
- [4] html5slides. <http://code.google.com/p/html5slides>.
- [5] Powered by mozilla. <http://www.mozilla.org/projects/powered-by.html>.
- [6] Telasocial digital signage with standards. <http://www.telasocial.com/index-en.html>.
- [7] B. B. Bederson, J. D. Hollan, J. Stewart, D. Rogers, A. Druin, and D. Vick. A zooming web browser. In *In 'Human Factors in Web Development*, 1999.
- [8] G. W. Furnas and B. B. Bederson. Space-scale diagrams: understanding multiscale interfaces. In *in Chi'95*, pages 234–241. ACM Press, 1995.
- [9] M. Galli. Tagvisor. <https://github.com/taboca/TagVisor>.
- [10] H. E. Hattab. Reveal.js. <http://lab.hakim.se/reveal-js>.
- [11] R. Heck, M. Wallick, and M. Gleicher. Towards virtual videography. In *Proc. Multimedia'00*, pages 375–378. ACM Press, 2000.
- [12] E. A. Meyer. S5: A simple standards-based slide show system. <http://meyerweb.com/eric/tools/s5>.
- [13] S. Pook, E. L. G. Vaysseix, and E. Barillot. Context and interaction in zoomable user interfaces. In *In Proceedings of the 5th International Working Conference on Advanced Visual Interfaces (AVI 2000)*, pages 227–231. ACM Press, 2000.
- [14] B. Szopka. Impress.js. <http://bartaz.github.com/impress.js>.

³TelaSocial experiments with interaction: <http://labs.telasocial.com/projects-interaction/>