Self-Contained Semantic Hypervideos Using Web Components

Thomas Steiner[1]*, Pierre-Antoine Champin[1],
Benoît Encelle[1], and Yannick Prié[2]

[1]CNRS, Université de Lyon, LIRIS – UMR5205, Université Lyon 1, France
{tsteiner, pierre-antoine.champin}@liris.cnrs.fr,
benoit.encelle@univ-lyon1.fr
[2]CNRS, Université de Nantes, LINA – UMR 6241, France
yannick.prie@univ-nantes.fr

Abstract. The creation of hypervideos—displayed video streams that contain embedded user-clickable anchors and annotations—is a manual and tedious job, requiring the preparation of assets like still frames, the segmentation of videos in scenes or chapters, and sometimes even the isolation of objects like faces within the video. In this paper, we propose a semi-automated Web-Components-based approach to self-contained hypervideo creation. By *self-contained* we mean that all necessary intrinsic components of the hypervideo, *e.g.*, still frames, should come from the video itself rather than be included as external assets. *Web Components* is a set of specifications, which let Web developers apply their HTML, CSS, and JavaScript knowledge to build widgets that can be reused easily and reliably. By leveraging this evolving standard, we obtain a high degree of abstraction, which reduces the burden of creating hypervideos to the familiar task of textually marking them up with HTML elements.

Keywords: Hypervideo, Web Components, semantic video annotation

1 Introduction

The term *hypervideo* is commonly used to refer to *"a displayed video stream that contains embedded user-clickable anchors"* [13,14] and annotations, allowing for navigation between the video and other hypermedia elements. In a 2006 article in *The Economist*, the authors write *"[h]yperlinking video involves the use of "object-tracking" software to make filmed objects, such as cars, clickable as they move around. Viewers can then click on items of interest in a video to watch a related clip; after it has played, the original video resumes where it left off. To inform viewers that a video is hyperlinked, editors can add highlights to moving images, use beeps as audible cues, or display still images from hyperlinked videos next to the clip that is currently playing"* [17]. In standard literature, hypervideo is considered a logical consequence of the related concept of *hypertext*.

* Second affiliation: Google, Hamburg, Germany, tomac@google.com

In contrast to hypertext, hypervideo necessarily includes a time component, as content changes over time. In consequence, hypervideo has different technical and aesthetic requirements than hypertext, the most obvious one being appropriate segmentation in scenes or even objects. The opportunities for feature-rich semantic hypervideos are endless, only limited by the feasibility and ease of their creation. With this paper, we propose an approach that is solely based on custom HTML elements that anyone coming from an HTML background can grasp.

2 Related Work

Related work can be regarded under the angles of online video annotation creation and large-scale Linked Data efforts for video. Many have combined Linked Data and video, typical examples are [7] by Lambert *et al.* and [6] by Hausenblas *et al.* There are several text track enriching approaches [9,10,11,15] based on named entity recognition. The online video hosting platform YouTube lets publishers add video annotations in a closed proprietary format. From 2009 to 2010, YouTube had a feature called Collaborative Annotations [1] that allowed video consumers to collaboratively create video annotations. In [19], Van Deursen *et al.* present a system that combines Media Fragments URI [18] and the Ontology for Media Resources [8] in an HTML5 Web application to convert media fragment annotations into a WebVTT file that can be used by HTML5-enabled players. Building on their work, in [16], we additionally allowed for writing annotations by letting annotators create WebVTT cues with an editor. Popcorn.js[1] is an HTML5 JavaScript media framework for the creation of media mixes by adding interactivity and context to videos by letting users link social media, feeds, visualizations, *etc.* to moving images. PopcornMaker[2] is an interactive Web authoring environment allowing for videos to be annotated on a timeline.

3 Web Components

Web Components is a set of specifications, which let Web developers leverage their HTML, CSS, and JavaScript knowledge to build widgets that can be reused easily and reliably.[3] According to a (recently discontinued) W3C Working Draft introductory document,[4] the component model for the Web ("Web Components") consists of five pieces:

Imports which defines how templates, decorators and custom elements are packaged and loaded as a resource [5].

Shadow DOM which encapsulates a DOM subtree for more reliable composition of user interface elements [4].

[1] Popcorn.js: `http://popcornjs.org/`

[2] PopcornMaker: `https://popcorn.webmaker.org/`

[3] Web Components: `http://www.chromium.org/blink/web-components`

[4] Discontinued W3C Working Draft document: `http://www.w3.org/TR/2013/WD-components-intro-20130606/` [2]

Custom Elements which let authors define their own elements, with new tag names and new script interfaces [3].
Decorators which apply templates based on CSS selectors to affect rich visual and behavioral changes to documents.
Templates which define chunks of inert markup that can be activated for use.

At time of writing, native support for Web Components has just landed in a number of Web browsers, however, for the majority of browsers, a so-called polyfill solution is still required. A polyfill is a piece of code (or plugin) that provides the technology that developers expect the browser to provide natively in the near future. We rely on the Polymer project[5] that provides Web Components support for older browsers. Polymer allows us to create reusable widgets that introduce a number of new custom HTML elements for our task of hypervideo creation.

4 Implementation Details

We have developed a number of Web Components for the creation of hypervideos. These Web Components are behaviorally grouped together by a common naming convention. In Polymer, all element names have to start with "polymer-".

<polymer-hypervideo> is the parent element of all other elements. It accepts the attributes `src` for specifying a set of space-separated video sources (to support different encodings), and—analog to the native HTML5 video attributes—`width` and `height` for specifying the video's dimensions, then `poster` for specifying the video's poster frame, and finally `muted` to specify if the video should be initially muted.
<polymer-data-*> is a set of data annotation elements that includes the two shorthand annotation types `<polymer-data-actor>` for annotating video actors and `<polymer-data-overlay>` for annotating visual overlays, and the generic `<polymer-data-annotation>` for other annotations.
<polymer-track-*> are the two elements `<polymer-track-chapters>` and `<polymer-track-subtitles>`, which internally rely on WebVTT [12] text tracks of the types "chapters" and "subtitles" that they enrich with automatically generated chapter thumbnails and a full text subtitle view.
<polymer-visualization-*> currently provides the following two visualization elements `<polymer-visualization-timeline>` on the one hand and `<polymer-visualization-toc>` on the other that create a timeline view and a table of contents that put all encountered `<polymer-track-*>` and `<polymer-data-*>` elements in a temporal context.

We have made an online demo application available at `http://hypervideo.herokuapp.com` and also share its source code.[6] As both native Web Component support in Web browsers and the Polymer project are constantly evolving and still in flux, the demo currently works best on Opera v22.0.1471.70

[5] Polymer project: `http://www.polymer-project.org/`
[6] Hypervideo Demo: `https://github.com/tomayac/postdoc/tree/master/demos/polymer-hypervideo`

(Mac OS X). A screenshot of the application can be seen in Figure 1, the corresponding underlying code sample is shown in Listing 1. These components communicate with each other through standard JavaScript events, so when a components needs to communicate its state to another, *e.g.*, `<polymer-hypervideo>` the current time of the video to one of the visualization components like the `<polymer-visualization-timeline>`, it fires an event that components can subscribe to and react upon. Listing 2 shows the relevant code snippets.

5 Linked Data Considerations

The *Frequently Asked Questions* section of the Polymer project states: "As Polymer makes use of polyfills, search engines treat Polymer-based applications no differently than they do other JavaScript-based Web apps. In fact, Google's crawler understands JavaScript heavy applications.[7] Going forward, it is a reasonable assumption that as use of native Shadow DOM [4] increases, search engine providers will try to adapt to understand it, just as they have adapted to

[7] Google Web application understanding: `http://googlewebmastercentral.blogspot.com/2014/05/understanding-web-pages-better.html`

```
<polymer-hypervideo src="big_buck_bunny.mp4,big_buck_bunny.webm" width="400" height="225" muted>
  <polymer-data-actor start="10" end="35" name="Chirp (Owl)" xywh="170,20,70,80"
      url="http://commons.m.wikimedia.org/wiki/File:Chirp1_-_BBB_-_reduced_snapshot.png">
  </polymer-data-actor>
  <polymer-track-subtitles src="subtitles.vtt" displaysubtitlesgroup></polymer-track-subtitles>
  <polymer-track-chapters src="thumbs.vtt" displaychaptersthumbnails></polymer-track-chapters>
  <polymer-visualization-timeline orientation="landscape"></polymer-visualization-timeline>
  <polymer-visualization-toc></polymer-visualization-toc>
</polymer-hypervideo>
```

Listing 1. Web Components mark-up for the hypervideo in Figure 1, including subtitles, chapters, timeline, and table of contents widgets; the actor annotation contains a spatial fragment (`xywh`) [18] and a hyperlink (`url`) to Wikimedia Commons

```
// === In polymer-hypervideo.js: ===
// listen to native html5 video timeupdate events
video.addEventListener('timeupdate', function() {
  that.currentTime = video.currentTime;
  // publish hypervideotimeupdate events
  that.fire('hypervideotimeupdate', { currentTime: that.currentTime });
});

// === In polymer-visualization-timeline.js: ===
// listen to hypervideotimeupdate events
document.addEventListener('hypervideotimeupdate', function(e) {
  var currentTime = e.detail.currentTime;
  // update the time marker
});
```

Listing 2. Native JavaScript event communication between Web Components

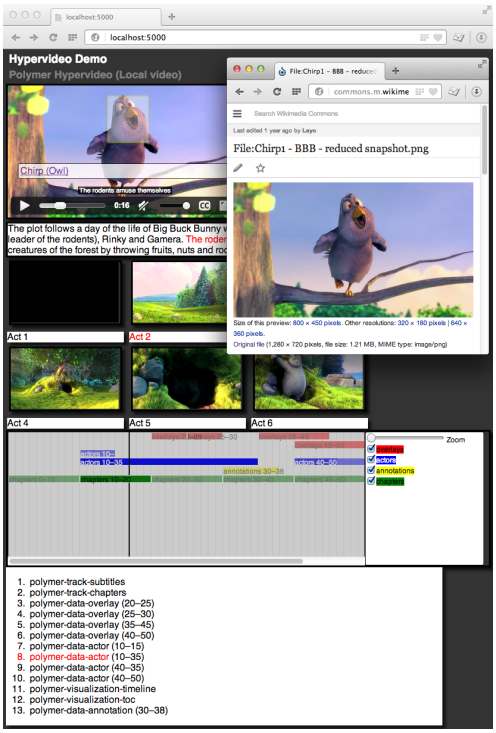

Fig. 1. Generated hypervideo based on the mark-up in Listing 1, including subtitles, chapters, timeline, and table of contents widgets; the actor annotation contains a spatial fragment [18] (owl's head) and a hyperlink to a details page on Wikimedia Commons

other new Web technologies in the past."[8] In the longterm, this paths the way toward the semantic information introduced through Web Components mark-up not being lost by search engines and Linked Data crawlers, but being understood.

6 Conclusions and Future Work

In this paper, we have introduced a Web-Components-based approach for the generation of hypervideos with nothing more than custom HTML elements. We have made both a demo application and the underlying source code as well as the source code of the mentioned Web Components available. Immediate next steps are to improve the browser compatibility and to migrate the code to the native Web Components implementation that will soon be encountered in all major Web browsers. Concluding, Web Components radically redefine how we develop applications on the Web. With our code, we have shown that this certainly holds true for hypervideo and are excited about future applications and use cases.

[8] Polymer FAQ: `http://www.polymer-project.org/resources/faq.html`

References

1. S. Bar et al. YouTube's Collaborative Annotations. In *Webcentives '09, 1st International Workshop on Motivation and Incentives*, pages 18–19, 2009.
2. D. Cooney and D. Glazkov. Introduction to Web Components. Working Draft, W3C, June 2013. http://www.w3.org/TR/components-intro/.
3. D. Glazkov. Custom Elements. Last Call Working Draft, W3C, Oct. 2013. http://www.w3.org/TR/custom-elements/.
4. D. Glazkov and H. Ito. Shadow DOM. Working Draft, W3C, June 2014. http://www.w3.org/TR/shadow-dom/.
5. D. Glazkov and H. Morrita. HTML Imports. Working Draft, W3C, Mar. 2014. http://www.w3.org/TR/html-imports/.
6. M. Hausenblas, R. Troncy, Y. Raimond, and T. Bürger. Interlinking Multimedia: How to Apply Linked Data Principles to Multimedia Fragments. In *Linked Data on the Web Workshop (LDOW 09), in conjunction with the 18th International World Wide Web Conference (WWW 09)*, 2009.
7. D. Lambert and H. Q. Yu. Linked Data based Video Annotation and Browsing for Distance Learning. In *SemHE '10: The Second International Workshop on Semantic Web Applications in Higher Education*, 2010.
8. W. Lee, W. Bailer, T. Bürger, et al. Ontology for Media Resources 1.0. Recommendation, W3C, Feb. 2012. http://www.w3.org/TR/mediaont-10/.
9. Y. Li, G. Rizzo, J. L. Redondo García, R. Troncy, M. Wald, and G. Wills. Enriching Media Fragments with Named Entities for Video Classification. In *Proceedings of the 22nd International Conference on World Wide Web Companion*, WWW '13 Companion, pages 469–476, 2013.
10. Y. Li, G. Rizzo, R. Troncy, M. Wald, and G. Wills. Creating Enriched YouTube Media Fragments with NERD Using Timed-Text. In *11th International Semantic Web Conference (ISWC2012)*, November 2012.
11. Y. Li, M. Wald, T. Omitola, N. Shadbolt, and G. Wills. Synote: Weaving Media Fragments and Linked Data. In C. Bizer, T. Heath, T. Berners-Lee, and M. Hausenblas, editors, *LDOW*, volume 937 of *CEUR Workshop Proceedings*, 2012.
12. S. Pfeiffer and I. Hickson. WebVTT: The Web Video Text Tracks Format. Draft Community Group Specification, W3C, Nov. 2013.
13. N. Sawhney, D. Balcom, and I. Smith. HyperCafe: Narrative and Aesthetic Properties of Hypervideo. In *Proceedings of the the Seventh ACM Conference on Hypertext*, HYPERTEXT '96, pages 1–10, New York, NY, USA, 1996. ACM.
14. J. Smith and D. Stotts. An Extensible Object Tracking Architecture for Hyperlinking In Real-time and Stored Video Streams. *Department of Computer Science, University of North Carolina at Chapel Hill, Technical Report*, 2002.
15. T. Steiner. In *Proceedings of the ISWC 2010 Posters & Demonstrations Track: Collected Abstracts, Shanghai, China, November 9, 2010.*
16. T. Steiner, H. Mühleisen, R. Verborgh, P.-A. Champin, B. Encelle, and Y. Prié. Weaving the Web(VTT) of Data. In *Proceedings of the 7th Workshop on Linked Data on the Web*, Apr. 2014.
17. The Economist. From Hypertext to Hypervideo, Sept. 2006. http://www.economist.com/node/7904166.
18. R. Troncy, E. Mannens, S. Pfeiffer, et al. Media Fragments URI 1.0 (basic). Recommendation, W3C, Sept. 2012. http://www.w3.org/TR/media-frags/.
19. D. Van Deursen, W. Van Lancker, E. Mannens, et al. Experiencing Standardized Media Fragment Annotations Within HTML5. *Multimedia Tools and Applications*, pages 1–20, 2012.