LADS Tour Authoring & Playback System

Master's Project Report

May 20, 2011

James C. Chin
jcchin@cs.brown.edu

Department of Computer Science
Brown University
Providence, RI 02912

CONTENTS

1 ABSTRACT

LADS (Large Artwork Display on the Surface) is an active Brown University project, sponsored by Microsoft Research, that explores innovative methods of touch-based exploration and interaction with high-resolution, large-format artwork and the context of that art. Its two target audiences are casual "walk-up" novices and more goal-directed, research-oriented scholars. "Walk-up" users, in particular, could use LADS in a museum setting to browse and explore artworks at their leisure and/or view a pre-constructed guided tour that takes the place of a docent or audio guide at an exhibition.

This report focuses on the latter feature, the LADS Tour Authoring & Playback System, which enables one to easily create a guided tour for a piece of artwork with an audio narration and various animations, including the panning/zooming of the artwork and the fading/panning/zooming of associated media connected to that piece of art. Such tours should greatly enhance the art museum viewing experience, as curators would be able to provide visitors with comprehensive, high-resolution overviews of large artwork in addition to the ability to explore and interact with that art though a touch-based interface.

2 INTRODUCTION

LADS (Large Artwork Display on the Surface) is an active Brown University project, sponsored by Microsoft Research, that explores innovative methods of touch-based exploration and interaction with high-resolution, large-format artwork and the context of that art. Its two target audiences are casual "walk-up" novices and more goal-directed, research-oriented scholars. "Walk-up" users, in particular, could use LADS in a museum setting to browse and explore artworks at their leisure and/or view a pre-constructed guided tour that takes the place of a docent or audio guide at an exhibition.

Indeed, many art museums only feature audio-based tours. For instance, a visitor might need to wear a pair of headphones connected to a portable audio playback device to listen to narrated tours as he or she explores different artworks throughout a particular art museum. However, an audio-only tour is limiting, as a visitor might have trouble figuring out the location of the artwork being referred to by a certain portion of the narration. In addition, the visitor is restricted to the scale of the artwork that he or she is able to see (or allowed to see). For instance, a painting might be so large that a visitor physically could not be tall enough to see the upper portion of the artwork, or visitors might be restricted to viewing an extremely valuable piece of art from a distance.

In recent years, museums in general have employed additional technology to enhance the visitor experience. Various platforms, including interactive guides and informative kiosks, have been used to engage visitors and enhance their understanding of exhibits. However, while the possibility of offering these types of experiences might be very appealing, the design and development of such multimedia applications often requires a great deal of time, money, and energy. For instance, a museum might need to outsource such development to an external firm that specializes in the creation of such installations. Even if a museum could create such applications internally, it might require staff with extensive technical backgrounds, as many of today's video/animation editing programs are quite complex.

To address this problem, this report presents the LADS Tour Authoring & Playback System, which enables one to quickly and easily create a guided tour for a piece of artwork with an audio narration and various animations, including the panning/zooming of the artwork and the fading/panning/zooming of associated media connected to that piece of art. Such tours should greatly enhance the art museum viewing experience, as curators (even those with only a basic technical background) would be able to provide visitors with comprehensive, high-resolution overviews of large artwork in addition to the ability to explore and interact with that art though a touch-based interface.

3 RELATED WORK

Microsoft Research's Project Tuva presents Nobel Prize-winning physicist Richard Feynman's Messenger Series lectures in an enhanced Silverlight-based video player. The seek bar for the close-captioned, searchable video player is split up into a number of bookmarked sections, and the bar is synchronized with the two tracks below it, "Notes" and "Extras." The "Notes" track allows a user to annotate a lecture with text-based notes at specified times during the video. These notes are stored in a "Notes" pane to the left of the video viewing area, so the user can click on each time-stamped note to jump to that point in a particular lecture. In addition, the "Extras" track is filled with icons of associated media (text, images, and external links). When the playhead moves over one of these icons, the "Extras" pane to the right of the video viewing area auto-scrolls downward to highlight the corresponding media item, which the user can examine while a lecture is paused. Finally, just above the video viewing area is a shaded commentary box that allows the user to select from a list of project contributors and load that person's text-based commentary, which is synchronized with a lecture's video.

Another piece of related work is Microsoft Research's WorldWide Telescope (WWT), which is a Silverlight-based tool that enables users to explore rich, specialized visualizations of astronomical and planetary data through maps, data sets, and animations. Specifically, it allows one to seamlessly pan and zoom across the night sky by blending terabytes of images, data, and stories from multiple sources over the Internet into an immersive, media-rich experience. Users can even customize WWT to get their own message across in several different ways, one of which is through guided tours. While there are a number of pre-made tours of the cosmos already accessible via the WWT client, users can easily author their own animated slide-based tours by using the creation and editing screens, menus, and dialogs. The user can add metadata, music, voiceovers, text, shapes, and images to a tour, which can then be saved as a single file and shared with others.

Finally, Microsoft Research's Rich Interactive Narratives (RIN) project aims to combine traditional forms of storytelling with new visualization technologies to create powerful interactive digital narratives. Typically, traditional narratives (like guided tours) have been passive experiences in which the viewer is limited to the path that the author has defined. However, RIN enables a viewer to experience a narrative and the underlying environment as a guided tour while allowing for deeper, context-specific, and open-ended exploration at any time. For instance, a viewer might wish to pause a tour to explore a painting, an interesting sculpture, or a different room that the tour's path does not explicitly cover. The RIN data model is a structured, extensible, and platform-independent representation for narratives that is efficient and robust. In addition to building a new Silverlight-based content player that enables a user to interact with and explore a narrative, the RIN authors have also built a new authoring tool that simplifies content creation through an intuitive WYSIWYG interface. The extensibility of RIN allows a user to incorporate present and future forms of rich Experience Streams – those currently available include visualizations like interactive maps via Bing, gigapixel images via Deep Zoom, and 3D scene reconstructions from photos via PhotoSynth. With these built-in visualization experiences, one can create a wide variety of compelling content without any programming.

4 DESIGN

First, we will discuss the basic workflow of the LADS Tour Authoring & Playback System. Then, we will dive into the backend architecture, which deals primarily with the structure and flow of the underlying data model. Finally, we will describe the authoring and playback UIs, including details about the visual layout and interaction design. This system was implemented in C# and XAML using Windows Presentation Foundation (WPF), the graphical subsystem that is used to render user interfaces in Surface-based applications.

4.1 Overall Workflow

First, the user selects a piece of artwork to view from the LADS Artwork Selection Mode. Then, the user selects a button in the "Tours" section of the left sidebar to enter the tour authoring mode. He or she is then presented with an initial dialog box and given the choice of either creating a new tour or loading an existing one from an XML file. Once the user chooses an option, he or she is presented with the tour authoring UI, which loads as a layer on top of the LADS Artwork Viewing Mode.

The top half of the tour authoring UI is the artwork/associated media viewing area, and the bottom half of the UI is the storyboard area where a track is present for each item (artwork, associated medium, or audio) that is part of the tour. Once the user is done editing a tour, he or she saves the tour session, thereby updating the tour definition defined in the underlying XML file in the process. Finally, this XML file can then be edited more through the authoring UI or loaded as a tour for playback for the LADS end user.

4.2 Backend Architecture

The backend architecture is centered on the LADS tour data model, which is present in three cohesive formats for storage, editing, and playback.

The storage format for a tour, an XML specification, starts off with the root TourStoryboard node that has attributes for the "displayName" and "description" of the tour. The root node contains either TourParallelTL or TourMediaTL nodes, which have direct counterparts in the editing and playback data formats. A TourParallelTL represents the timeline of animations for a particular tour item, whose type ("artwork" or "media") is represented as a "type" attribute. A TourParallelTL node also has attributes for the item's "displayName" and "file" (the name of the corresponding source file). Each TourParallelTL node contains children TourEvent nodes that are defined by their "beginTime," "type," and the properties associated with a particular TourEvent type (ZoomMSIEvent (panning/zooming an artwork), FadeInMediaEvent, ZoomMediaEvent, or FadeOutMediaEvent). For instance, a ZoomMediaEvent has the following properties: "toScreenPointX," "toScreenPointY," "scale," and "duration." A TourMediaTL corresponds to a MediaTimeline in WPF, and, like a TourParallelTL, it simply contains attributes for its "type" ("audio"), "displayName," and MP3 "file."

When a tour is loaded from an XML file, the specification is converted to a live, intermediate dictionary-based data model that can be edited, converted to the playback format, or saved as an XML file once again. In other words, when a tour is being edited, this is where the changes actually occur. The top-level "tourDict" dictionary, which corresponds to the TourStoryboard XML node, contains elements where the key is a WPF Timeline and the value is a dictionary of TourEvents keyed by their "beginTimes." These TourParallelTL and TourMediaTL sub-dictionaries correspond to their XML counterparts, and they have been extended from their WPF parent classes (ParallelTimeline and MediaTimeline) to store the additional attributes of "type," "displayName," and "file." (Of course, a TourMediaTL does not contain any TourEvents.)

Finally, the playback format is built on the WPF Storyboard class, which is ideal for controlling multiple timelines of animations. When a tour is ready to be played (either while one is authoring or viewing a tour), the appropriate animations for each TourEvent are created and added to their respective TourParallelTLs. (Again, a TourMediaTL does not contain any TourEvents.) Then those timelines are added to the overall "tourStoryboard" to prepare it for playback.

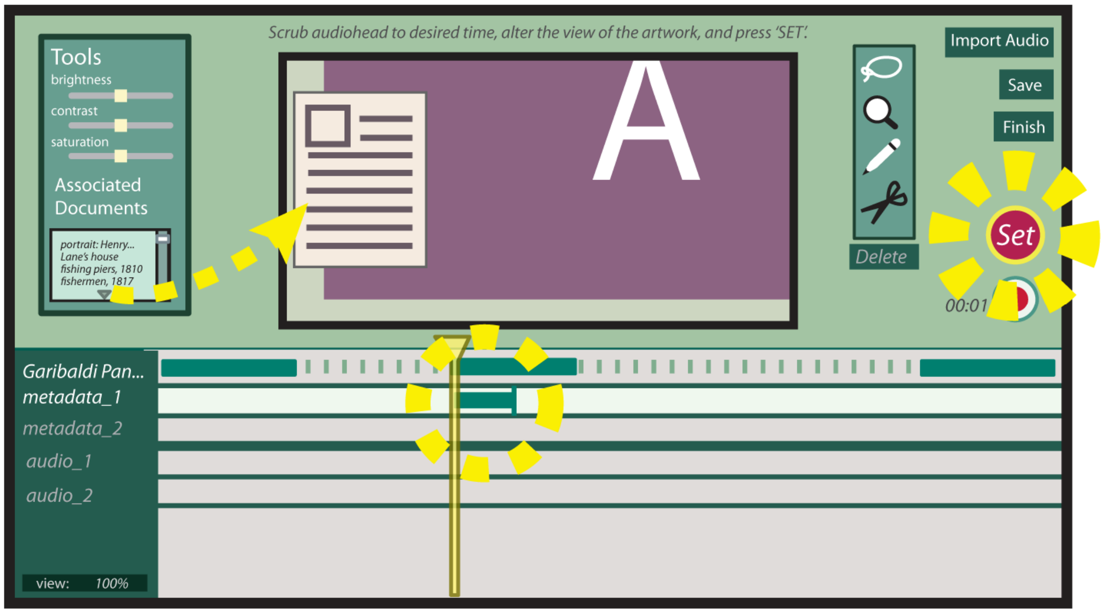

Figure 1: April 2011 mockup of authoring UI, which has since been pared down to meet the LADS 1.0 release deadline of late May 2011.

4.3 Authoring UI

As stated in the "Basic Workflow" section above, the user selects a piece of artwork to view from the LADS Artwork Selection Mode. Then, the user selects a button in the "Tours" section of the left sidebar to enter the tour authoring mode. He or she is then presented with an initial dialog box and given the choice of either creating a new tour (entering a name and description) or loading an existing one from an XML file. Once the user chooses an option, he or she is presented with the tour authoring UI, which loads as a layer on top of the LADS Artwork Viewing Mode. The top half of the tour authoring UI is the artwork/associated media viewing area, and the bottom half of the UI is the storyboard area where a track is present for each item (artwork, associated medium, or audio) that is part of the tour.

The column in the storyboard area to the left of the timelines is for the timeline metadata. Just above the top of this column will be the "displayName" for the tour, and the rows of this column will contain the "displayName" for each corresponding timeline. To the left of each timeline "displayName" will be an icon identifying the timeline's type ("artwork," "media," or "audio"). The timeline area itself can be panned and zoomed, and a ruler along the top of the timeline area will change accordingly. While authoring a tour, the user can seek to any point along the timeline ruler by touching the yellow playhead handle on top of the timeline area, dragging the handle to the desired time, and letting go of the handle. The user can also preview all or part of a tour at any point by using the "Play" and "Pause" buttons located in the upper right-hand corner of the screen. The operations that can be done on the overall tour structure (list of timelines) are as follows:

- Add timeline:
  - The artwork timeline will be the default one that comes with a new tour (the user cannot delete this timeline).
  - The media timelines will be added based on what the user loads from the "associated media" box in the left sidebar, which is initially collapsed upon entering the tour authoring UI mode.
  - Create the audio timeline by tapping the "Import MP3" button on the bottom of the metadata column to import an MP3 via a dialog box (only one audio timeline allowed).
- Delete timeline: tap a timeline's title area to select a timeline (it will be outlined in red). Then tap the "Delete" button on the bottom of the metadata column to remove the timeline (the user is not allowed to delete the artwork one).

Each TourEvent within an artwork or media timeline is represented by a TourEvent bar. The event's "beginTime" corresponds to the position of the left side of the bar with respect to the timeline ruler, and the length of the bar represents the event's "duration." (Within a timeline, TourEvents are not allowed to overlap.) Manipulations of these bars will change the properties of the underlying TourEvent, reload the "tourDict" to the "tourStoryboard," and seek the "tourStoryboard" back to the beginning of the TourEvent that was changed. The operations that can be done on a TourEvent bar are as follows:

- Add TourEvent:
  - Select a timeline (it will be outlined in red).
  - Drag the playhead to the position where you want the TourEvent to begin.
  - Pan/zoom the timeline item (artwork or associated medium) to the desired "end position/scale" (irrelevant for FadeOutMediaEvent).
  - On the right side of the screen, just above the storyboard, tap the desired TourEvent-typed add button ("New ZoomMSIEvent" for artwork, "New FadeInMediaEvent"/"New FadeOutMediaEvent"/"New ZoomMediaEvent" for media). A default 1-second TourEvent will be created.
- Modify TourEvent:
  - Modify "beginTime": touch an existing TourEvent bar to highlight it (its color will become a bit brighter) and make it the active one (the playhead jumps to the beginning of the bar). Then drag it left or right along its timeline to change the TourEvent's "beginTime," which corresponds to the position of the left side of the bar with respect to the timeline scale at the top of the timeline area. Finally, tap

    the “Save TourEvent Changes” button on the right side of the screen, just above the storyboard.
  - Modify “duration”: touch an existing TourEvent bar to highlight it (its color will become a bit brighter) and make it the active one (the playhead jumps to the beginning of the bar). Then use a two-finger pinching gesture to modify the event’s duration (the right side of the bar expands and contracts while the left side stays anchored to its current position). Finally, tap the “Save TourEvent Changes” button on the right side of the screen, just above the storyboard.
  - Modify end position/scale: touch an existing TourEvent bar to highlight it (its color will become a bit brighter) and make it the active one (the playhead jumps to the beginning of the bar). (If the event is a FadeInMediaEvent, the medium become visible for editing.) Pan/zoom the timeline item (artwork or associated medium) to the desired “end position/scale” (irrelevant for FadeOutMediaEvent). Finally, tap the “Save TourEvent Changes” button on the right side of the screen, just above the storyboard.
- Delete TourEvent: touch an existing TourEvent bar to highlight it and make it the active one (the playhead jumps to the beginning of the bar). Then tap the “Delete TourEvent” button on the right side of the screen, just above the storyboard.

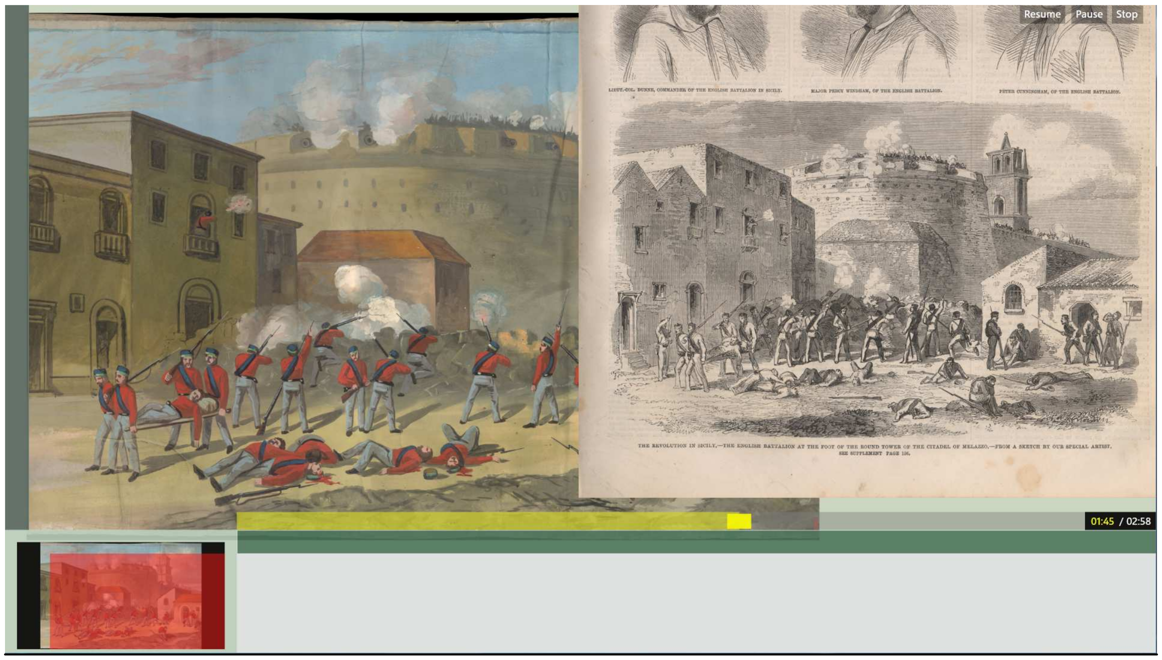

Figure 2: Screenshot of playback UI that shows a still from the author's tour of "English battalion at the foot of the round tower, Citadel of Milazzo," a scene from J.J. Story's *Garibaldi Panorama* (1860).

4.4 Playback UI

When a tour is loaded for playback, the left sidebar is toggled to the "closed" position to allow the user to see more of the artwork. In addition, the LADS Artwork Viewing Mode buttons in the upper right-hand corner are replaced with "Resume," "Pause," and "Stop" buttons for controlling the tour. A translucent gray seek bar also shows up right above the workspace area in the bottom area of the screen, and at the end of this bar on the right is an area that displays the progress of the tour in terms of [elapsed time]/[total time]. Finally, the artwork viewing area is replaced with the tour viewing area.

As the tour plays, a yellow playhead moves across the seek bar from left to right, and the portion that has been played so far is filled with a translucent yellow color. The user can seek to any point along the seek bar by touching the playhead, dragging it to the desired time, and letting go of the playhead. When the tour stops (either by itself or via the "Stop" button), the left sidebar is toggled open again, the tour control buttons in the upper-right hand corner are replaced with LADS Artwork Viewing Mode navigation buttons, the seek bar disappears, and the artwork viewing area is returned to the state that it was in before it was swapped with the tour viewing area.

5 DISCUSSION

5.1 Challenges

At first, the WPF Storyboard class was not used as the data model for playback, as initial prototypes simply used a basic .NET DispatcherTimer to fire off TourEvents at specified "beginTimes." However, the DispatcherTimer class was found to be unreliable, as animations could momentarily stop the timer from ticking. Also, this primitive data model did not allow one to easily implement controls for pausing/resuming a tour or seeking within a tour. In fact, after some careful thought, it soon became clear to the author that it would be quite infeasible to attempt to implement such controls while using the DispatcherTimer class. Fortunately, the author discovered and used the WPF Storyboard class, which takes care of all of these issues.

Another roadblock was the issue of how to get the duration of the entire "tourStoryboard." The default duration of a Storyboard in WPF is "Automatic," which means that the Storyboard will expand to fit child timelines and playback will not end until the last child timeline stops playing. However, for the implementation of the seek bar and the "total time" portion of the time display area, a specific, finite "tourStoryboard" duration (in numbers) is needed – "Automatic" does not work. It turns out that the duration of a Storyboard, when not specified by the end user, is calculated internally upon playback, and there is no way to extract that value for programming purposes. Thus, the duration of the "tourStoryboard" is now calculated and specified during the conversion of the "tourDict" into the "tourStoryboard" for playback. During this loading process, the latest end time ("beginTime" + "duration") of any TourEvent is tracked until the maximum value is found (it could be the duration of the TourMediaTL MP3 file). This final value is then set as the duration of the "tourStoryboard."

Speaking of the duration of the TourMediaTL MP3 file, WPF also has no way of figuring out the duration of the audio in the file. The MediaElement class (used to play audio and video within a MediaTimeline class) does provide a property called NaturalDuration. However, this is only calculated after the MediaElement is loaded, an event that is not fired until the MediaElement is called to start playing. Fortunately, there is a fairly popular open-source package available online that allows one to easily read and edit the metadata of several popular audio formats – it is called "taglib-sharp," and it serves it purpose quite well.

5.2 Future Work

In the future, the LADS Tour Authoring & Playback System could be enhanced to include more advanced TourEvents and more functionality. For instance, one possible TourEvent could allow the user to highlight the compositional overlay of an artwork and animate the paths of such an overlay during a tour. In addition, another possible TourEvent could highlight areas of an artwork by applying a black opaque mask around those areas to isolate them. Yet another possible TourEvent could overlay magnified areas of an artwork in place – this would resemble the effect of taking snapshots of certain areas and panning/scaling those snapshots as overlays on the top of the artwork.

Meanwhile, additional functionality could include adding the ability to record audio segments for the narration and to do basic audio editing (changing “beginTime” and “duration”). Also, another additional feature could be the ability to interact with the artwork/associated media when a tour is paused (similar to what Microsoft’s Rich Interactive Narratives project allows a user to do). Finally, a simple spline editor could be added to provide a tour author with more fine control over TourEvent animations, but this might be too extraneous.

6 CONCLUSION

The LADS Tour Authoring & Playback System is an integral feature of the LADS (Large Artwork Display for the Surface) project as a whole. At its core, the system simply enables a user to author a tour by adding artwork/associated media to a storyboard, configuring animations of those items to run at certain times, and importing an audio narration file in MP3 format. However, the development of such a system is actually quite complex, as there are many moving parts that need to be integrated together, especially those of the user interface layer and the backend architecture. Nevertheless, one of the goals of this project is to make the system easy enough to use for the average museum curator, and the design of the system certainly strives to accomplish that, despite the implementation challenges and limitations of working with Microsoft's Windows Presentation Foundation system, which is required for Surface development. In the end, this system should enhance one's artwork viewing and learning experience for the better, whether LADS is deployed in a museum or elsewhere.

7 REFERENCES

About Project Tuva - Microsoft Research. http://research.microsoft.com/en-us/collaboration/focus/education/tuva.aspx.

Elisa Rubegni, Amalia Sabiescu, and Paolo Paolini. 2008. OneThousandAndOneStories: a format for multichannel multimedia narratives. In *Proceedings of the 15th European conference on Cognitive ergonomics: the ergonomics of cool interaction* (ECCE '08), Julio Abascal, Inmaculada Fajardo, and Ian Oakley (Eds.). ACM, New York, NY, USA, Article 19, 2 pages.

Elisa Rubegni, Nicoletta Di Blas, Paolo Paolini, and Amalia Sabiescu. 2010. A format to design narrative multimedia applications for cultural heritage communication. In *Proceedings of the 2010 ACM Symposium on Applied Computing* (SAC '10). ACM, New York, NY, USA, 1238-1239.

Microsoft Research Digital Narratives. http://www.digitalnarratives.net/.

Neeharika Adabala, Naren Datha, Joseph Joy, Chinmay Kulkarni, Ajay Manchepalli, Aditya Sankar, and Rebecca Walton. An interactive multimedia framework for digital heritage narratives. In *Proceedings of the international conference on Multimedia* (MM '10). ACM, New York, NY, USA, 1445-1448.

Project Tuva: Enhanced Video Player Home - Microsoft Research. http://research.microsoft.com/apps/tools/tuva/index.html.

Rich Interactive Narratives - Microsoft Research. http://research.microsoft.com/en-us/projects/rin/.

WorldWide Telescope. http://www.worldwidetelescope.org/.

WorldWide Telescope - Microsoft Research. http://research.microsoft.com/en-us/projects/wwt/.