LonghornLink:
Mapping RSS Feeds to Visualize a College in Action
Lawrence Archer, April DeRome, Seth C. Lewis, Ji Hyun Park
University of Texas at Austin
School of Information
1 University Station D7000
Austin, TX 78712
+1 (512) 471-3821
{larcher, aderome, seth.lewis, jhpark}@mail.utexas.edu

ABSTRACT
Our mash-up project, LonghornLink, seeks to resolve the limitations of conventional calendars and current mapping options by providing a new geographical and temporal representation of events at the University of Texas at Austin (UT). Combining dynamic event data via RSS feeds with static data on campus locations and sites of interest, LonghornLink allows users to gain more contextual knowledge—geospatially and temporally—about activities. As such, it offers a first, tentative step toward creating a “one-stop shop” interface for visualizing the UT campus.

Categories and Subject Descriptors
H.5.4 [Hypertext/Hypermedia]
H.4.3 [Communication Applications]: Information Browsers.

General Terms
Documentation, Design, Human Factors, Standardization

Keywords
RSS, Google Maps API, geospatial tagging, mapping.

1. INTRODUCTION
Calendars provide answers concerning what an event is about, when it is to occur, and where it is to take place. This information usually is provided textually, limiting how a user can interact with the event data. Data visualization can transform data into graphical representations allowing for new formats to be explored. By combining calendar data with a geographical interface (Google Maps), our project provides a fresh interface for calendars, taking advantage of graphical representation to better answer the where and when questions regarding events. Our project, called LonghornLink (http://voyager.ischool.utexas.edu/~larcher/map), proposes a mash-up combining the OnCampus Events Calendar’s RSS feed with Google Maps to provide a dynamic mapping interface—one that facilitates users’ knowledge of, and thus likely involvement in, events on the main campus at the University of Texas at Austin (UT).

Using the JavaScript-based tools of the Google Maps application programming interface (API), along with continuously updated calendar information from UT’s Really Simple Syndication (RSS) feeds, we are able to embed not only dynamic, event-oriented data,
but also create a geospatial visualization of static data—building locations, departmental offices, and more.

The current OnCampus Events Calendar is the most comprehensive source of information about daily events occurring on the UT campus. Events are sorted into six categories (Arts and Humanities, Lifestyle and Community, Politics and International, Science and Technology, Sports and Health, and Business and Law) and can be viewed separately by topic or grouped together for a holistic view of all major activities on campus. The standard, short description for each event includes the time, description, and location. A more complete description for the event provides time, description, location, URL, contact, sponsor, admission, and category. In some instances the “location” field will link to a separate Web portal that has the UT campus building detail map. This feature, however, is not consistent with every event’s “location” listing.

Moreover, there are other problems with this approach. First, clicking on the “location” link takes the user away from the OnCampus Events Calendar page—requiring continuous backtracking if one would like to view the locations of several events. Additionally, the detailed maps of UT buildings are narrowly focused on particular campus “regions,” such that a user cannot visualize the individual building within the campus as a whole (see Figure 1). Meanwhile, a related challenge is that users seeking to visualize campus buildings in the broader context of campus as a whole are constrained, though in other ways, by Google Maps’ current representation of UT’s campus (see Figure 2). At the map-level view, the streets within campus but not the buildings themselves are visible in Google Maps’ portrayal, leaving users with an incomplete rendering—let alone a dynamic interface in which to explore additional details about sites around campus and the events going on within them.

Furthermore, in addition to helping users better visualize where activities take place, our project seeks to overcome the shortcoming of traditional calendars: the inability to visualize the temporal context for events. LonghornLink incorporates a SIMILE timeline to illustrate the chronological context in which events occur, offering users another avenue of data visualization.

Figure 2. Google Maps’ current representation of the UT campus (top), as compared to the LonghornLink mash-up that overlays the official campus map (above).

Figure 3. An overview perspective of the LonghornLink mash-up site (http://voyager.ischool.utexas.edu/~larcher/map).
Thus, to resolve these problems with a more user-oriented mapping interface that mashes static and dynamic data of interest to the UT community, we introduce LonghornLink (see Figure 3). This project overlays the overall detailed UT campus map onto Google Maps, creating a more thorough representation of campus that includes semantic zooming for macro and micro perspectives. Additionally, LonghornLink provides interactive functions that allow users to better learn about and explore campus events in their geographic and temporal context—to see not only what is happening where and when, but also learn about what to eat and where to park while there, among other practical information. Specifically, our map mash-up would offer users several layers of data visualization: (1) The map itself in the center of the page; (2) event listings on the left-hand side; (3) a timeline of events below the map; and (4) search (Figure 4) and filter (Figure 5) functions to find specific locations and learn more about nearby food and parking options.

In sum, our mash-up mapping project, LonghornLink, provides a new geographical and temporal representation of events at UT and additional information of interest. It combines static and dynamic data to go beyond merely answering the where and when of conventional calendars to provide crucial context—such as that achieved through semantic zooming in our Google Maps mash-up interface. Given that UT’s official campus maps—including an overall campus map [14], an accessibility map [13], an interactive Flash map [15], parking maps [11], and a shuttle-route map [16]—exist on different and sometimes hard-to-find Web pages, LonghornLink arrives as a first step toward creating a “one-stop shop” interface for visualizing the dynamic campus that is the University of Texas at Austin.

2. RELEVANCE

In this section, we briefly explain the concepts at work and place our project within the context of related work on mapping mash-ups and data visualization.

2.1 Mapping

Google Maps [4] distributes information in a spatial framework to provide context [6]. LonghornLink allows users to learn more about campus events by opening a window that provides relevant details, such as the departments located in the building, a photo of the building, and nearby buildings, food, and parking locations. This format allows users to explore campus in an interactive, dynamic context. In this way, a digital, interactive map can enhance information retrieval by providing a natural extension for the user [6].

Longhorn Link is a community-oriented map, concerned with a single geographic location and with a normative goal of enhancing knowledge about—and thus encouraging engagement in—community events at UT. Geller [3] has noted that community-driven maps are created by people who hold neither corporate nor governmental interests; these creators are usually people interested in mashing disparate information to provide a new informational representation.

2.2 Mash-ups

Mash-ups are Web applications that combine information from varied sources into a new format to showcase information in a novel and underrepresented way. They allow users to interact with once-disconnected data through fresh interfaces. Public APIs,
made available by several major Web services, have made it possible for Web developers to create mash-ups [20]. In 2008, an online survey revealed that Google Maps was the most commonly used API by mash-up developers. The survey data were not able to pinpoint why maps were most predominately used when creating mash-ups, except that maps were the most visual and that location data is easily extracted for mapping purposes [20]. As of this writing (April 26, 2008), Programmable Web lists 1,392 mash-ups using the Google Maps API [10].

### 2.3 RSS

RSS presents information in a machine-readable format, which allows for it to be automatically processed and filtered for easy consumption on a wide scale. Hansen et al. [5] call RSS a “family” of XML file formats that enable client programs to automatically check for changes or even combine RSS feeds from disparate sources into new Web pages. Such feeds are one possible data source for mash-ups like ours. For LonghornLink, we used the OnCampus Events Calendar page’s RSS feeds (one each for the six categories mentioned above) to populate our map with event information according to their geographic and temporal context.

Our mapping of RSS feeds resonates with work being done elsewhere. Micheal Young, of The New York Times R&D department, has built a mash-up that uses the Associated Press national news RSS feed to plot recent news on a Google Map; to geolocate each AP story, the Yahoo! Geocode API is used to pinpoint the longitude/latitude locations [19]. While using Google Maps is a popular choice, other mash-ups have been created using Yahoo! Maps [18] as the underlying mapping platform: The Salt Lake Tribune has created a mash-up using news stories that occur within seven miles of Salt Lake City. In both of these examples, longitude/latitude locations were easily accessible from the mash-up data; by contrast, for LonghornLink, finding these geolocation points for campus buildings proved difficult because of the lack of availability of such data, as will be explained further below.

### 2.4 Literature Review

Other researchers have used university campus maps to create similar mash-ups. In mapping the Portland State University campus, Murthy et al. [7] developed Mash-o-matic, a system that automatically extracts information from various sources and reassembles it in a map mash-up that, like our project, uses filters and superimposed information displayed via markers. Moreover, Murthy et al. [7] detailed problems of matching map coordinates to campus directories when only street addresses are provided—a challenge we have confronted already—and discusses some workarounds using geocoding data available online.

At another university, Connecticut State, researchers used Google Maps API to provide visual information about the geospatial distribution of students in a computer science program [9]. This was done by creating databases and XML scripting for the latitude-longitude coordinates for each student’s home zip code as well as the frequency of students per zip code. The result was a color-coded representation that highlighted clusterings of students in and around southern Connecticut. A key component in creating mash-ups involves overlaying existing maps onto Google Maps (or other Web mapping services). MapCruncher was conceived as a tool to help users more seamlessly accomplish this, rendering formerly static maps as dynamic and interactive (Elson). While not using this particular tool, LonghornLink accomplishes the same thing, taking the static UT campus map and overlaying it onto Google Maps such that data visualization and semantic zooming can be achieved in an interactive, click-and-drag format.

In contrast to these mash-ups of static data, researchers at AT&T Labs developed GeoTracker, a system that assembles and repurposes dynamic data—RSS feeds—to represent news and media in both a geospatial and chronological fashion, helping users grasp what is most recent and relevant to them [1]. This research connects with our project’s goal of geomapping current events to provide users with a more visual (and temporal) representation of campus activities. Using Google Maps API and Javascript, Chen et al. [1] explained how to mine RSS text to add geotags so that the latest feeds can be coordinated to specific map locations—an essential component in our proposed system.

Similarly, Pan and Mitra [8] built a system for visualizing information taken from text documents. Called FemarRepViz, this toolkit extracted and displayed temporal, geospatial and summarized information from FEMA National Update Reports. This work’s key contribution is its architecture for geotagging extracted data, such as person names and location names; in our project, we designed a mechanism for recognizing and tagging the names (and nicknames) of buildings, among other features.

Additionally, an important area of study has focused on end-user programming of mash-ups. While the data for mash-ups might be readily available, the process tends to require a high level of programming expertise. Tools such as Marmite [17] and Mashmaker [2] have been created to alleviate the need for intense programming so that less-skilled Web users can generate mash-ups. These studies highlight an increasing need for and interest in mash-ups, particularly mapping mash-ups. LonghornLink builds on this interest by providing users a dynamic, interactive interface through which to explore the UT campus.

### 3. SPECIFICATIONS

LonghornLink is a Web-based application and was constructed in a relatively short time frame. It was developed using JavaScript and several JavaScript APIs and libraries, and written and tested for the Mozilla Firefox browser. The interface is a single Web page containing map, event list, timeline, and various filters and controls.

The prototype attempts to be somewhat object-oriented and was developed with an eye towards future extensibility. However, as a prototype, there is room for improvement. Some aspects of the application could be more effectively implemented as server-side programming; future iterations would involve at least a partial rewrite of the code.

#### 3.1 Platform

The application is written in client-side JavaScript. It uses several freely available JavaScript APIs, including Google Maps, Google Feeds, SIMILE Timeline, and the jQuery Library (see Figure 6).

We chose a JavaScript-only approach (rather than a combination of client- and server-side processing) because it simplified the requirements for developing a prototype application. It allowed
for rapid and simple development of a Web application, and eliminated the need for a server, database configuration, nor any code to communicate with a database.

Future iterations of this application, however, will require a more substantial platform to deal with both a larger data set and an increase in users. For example, in the current JavaScript-only implementation, the application fetches and parses the campus event feed every time the page loads. An implementation of the program involving communication with a Web server and database would enable caching of this information, reducing page load and processing time across all users. More extensive use of Asynchronous JavaScript and XML (AJAX) technology would also decrease initial load time by allowing the client-side code to only retrieve and process the relevant event and location data from a server. The current prototype contains location data and descriptions for all buildings in an array of JavaScript objects that is processed with every page load.

3.2 Functionality

The program uses a map and a timeline to present a visualization of the events happening on the UT campus. The central object of the page is a Google Maps view of the UT campus area, with an image overlay to indicate the location of individual campus buildings. The overlay was built from the standard campus map images available from UT [14].

The left side of the page displays a list of campus events collected from the university’s event feeds, separated into the six aforementioned categories, with the ability to sort according to the time frame: today only, this week, this weekend, this month. When the page first loads, the program uses the Google Feeds API to fetch one of the RSS feeds of campus events. The feed

Figure 6. Map and feed initialization code.

function loadfeed( eventfeed ) {
    var feed = new google.feeds.Feed( eventfeed.URL );
    feed.load( function() {
       readEntries( feed.entries );
    } );
}

function readEntries( entries ) {
    if (entries.length) {
        var events = google.feeds.getEntriesByTagName( "events" );
        if (events.length) {
            // events fetched
        } else {
            // no events found
            return;
        }
    } else {
        // no feed found
        alert("No feed found.");
    }
}

function initMap() {
    if (google.maps.isBrowserCompatible) {
        var map = new google.maps.Map( document.getElementById( "map" ) );
        var bounds = new google.maps.LatLngBounds( new google.maps.LatLng(50.83, -114.09), new google.maps.LatLng(51.45, -124.83) );
        map.setMapCenter( new google.maps.LatLng( 37.81, -122.41 ) );
        // bounds for the map
        map.setMapCenter( new google.maps.LatLng( 37.81, -122.41 ) );
    }
}

Figure 7. From the pop-up windows, users can select tabs of interest: Building, Food, and Event Info.
information is parsed and stored in an array of JavaScript objects. There are two object classes in the prototype implementation of LonghornLink, corresponding to the two main types of data used: CampusEvent objects for encapsulating event data, and CampusPlace objects for location data. Each building on campus is described by a CampusPlace object. Each CampusEvent object contains the title, description, location, date, and time of the event, along with the text describing the event. The event's location was usually at the beginning of the text, but the limited time frame of the project forced us to find a more straightforward method of locating events.

The UT event data are also provided in iCal format (RFC 2445), in which the VEVENT component that describes an individual event has a discrete Location property. While the Google Feeds API does not natively deal with the iCal format, these calendars can be imported into Google Calendar, which can then export an Atom feed of the event data. This feed is then fetched and parsed by the Google Feeds API, with the Location field intact. From there, the location string is compared with a static list of building names to locate each event on the campus map. While converting between calendar and feed formats required some initial setup, it eliminated some of the need for heuristics in matching locations, or for identifying the location text within the entire event description. However, the lack of a standardized event vocabulary still necessitated the use of an alias list for the buildings.

3.1 Datasets

As described above, there are two types of data required for LonghornLink: event data and location data. Event data is collected from a UT Web site [12] that publishes RSS and iCal feeds of campus activities. Although the Google Feeds API provides a method for fetching and parsing RSS and Atom data, the format of the event data available through the campus RSS feed presented some difficulties for parsing. The event's location is included in the RSS “description” element (all of which is extracted from the RSS feed). The CampusEvent object stores the event location as a reference to a CampusPlace object (see Figures 8 and 9). As each CampusEvent object is created from the RSS data, the Location field of each RSS entry is parsed to assign the correct CampusPlace to the event. The string found in the Location field is compared against a set of known location strings, including official building names, building codes (three letters), and building “aliases” (other names for each building, including shortened forms, nick names, or abbreviations). If a match for a known building is found, a reference to the corresponding CampusPlace object is assigned to that CampusEvent.

Once the event feed is parsed and stored in the local array, it is sorted by event time and date (instead of by the date on which the feed item was published) and displayed in list format on the left (see Figure 3).

After the page loads, a user can browse through the list of events on the left. If the program found a matching location, clicking on the event will center the map on that location and open a pop-up window with information about the building. From this pop-up window, the user can select tabs that have relevant and related information, such as the location of nearby buildings, parking garages, and eateries (see Figure 7).

Finally, the program allows users to explore campus by searching for a specific building. Entering a building name (either its official name, its nickname, or its three-letter code) in the search box on the right will search the map for the building and highlight its location (see Figure 4).
enhancement to this application might include expanding the available layers of campus information to include other useful features, such as mailboxes, book drop-off boxes, parking lots, and bike racks.

4. EVALUATION
In this section, we will both describe preliminary testing we have conducted and consider how a more thorough evaluation could enhance the usability of our interface.

4.1 Preliminary Testing
Using a convenience sample of UT students and faculty, we explored qualitatively the general usefulness and ease of use of the mash-up map. Through an informal e-mail survey, these initial testers were asked about specific site features and potential add-ons:

1. Would having access to this kind of map interest you? If not, what could be changed to make it more useful?
2. What would you want to visualize from a map like this?
3. Where would be the most convenient places on the page for buttons, filters, etc.?
4. In the pop-up markers on the map, what information would you like to know (e.g., street-view pictures of buildings)?
5. Overall, what features would you most want to see?

In general, testers appreciated the ability to navigate campus in a novel way. However, they seemed to want a more personalized, customized approach—one that allowed them to answer “Where do I go?” questions about specific needs such as obtaining a campus parking pass. Additionally, with regards to the RSS feeds, they wanted a site that allowed them to conduct topic-specific searches for current and upcoming events related, for instance, to “engineering” or “journalism.” In other words, the point was to go beyond the standardized and rather simple who-what-when information provided about campus events—dynamic data itself wasn’t enough; it needed to have particular applicability and connect a one-time event to a larger world of information. As one respondent said: “Color-coded events depending on whether they’re music, lectures, conferences, plays, etc. would really make it easier to wade through the info. Taking it one step further, I usually end up checking Pandora or Hypem.com to listen to an artist, or checking Wikipedia to learn more about a visiting lecturer or topic being covered. Some kind of news feed, or another way to link to more information, would be great to have inside that popup bubble.”

In summary, one of the themes that emerged from preliminary testing was that users desired to “see” things not readily pictured on the map. That is, while they universally appreciated the ability to see campus events in the context of their geographic and temporal location, these respondents wished to put the daily stream of events and activities in the context of their personal location—their idiosyncratic interests, their major of study, and so forth.

4.2 Future Testing
A more thorough follow-up evaluation would seek to examine empirically the degree to which LonghornLink improves the gathering of information about campus activities and locations. We might consider exploring the following: (1) How many pages must be navigated to accomplish a task (e.g., becoming informed about a particular event)? (2) How long does it take to accomplish the task? (3) How readily do users remember information they have gained from the respective sites?

Our purpose would be to demonstrate that LonghornLink makes information about campus events—and surrounding amenities such as food and parking—more easily and quickly accessible. We would expect to find greater knowledge acquisition, enhanced satisfaction, and increased likelihood of attending campus events—all of which could be measured using Likert-scale surveys. Additionally, the site’s effectiveness could be tested in an experimental setting such as the on-campus Usability Lab, where more precise metrics could be studied.

5. CONCLUSION
Our mash-up project, LonghornLink, seeks to resolve the limitations of conventional calendars and current mapping options by providing a new geographical and temporal representation of events at the University of Texas at Austin (UT). Combining dynamic event data via RSS feeds with static data on campus locations and sites of interest, LonghornLink allows users to gain more contextual knowledge—geospatially and temporally—about activities. As such, it offers a first, tentative step toward creating a “one-stop shop” interface for visualizing the UT campus.

5.1 Future Applications
As mentioned previously, additional features could be added over time to enhance the usefulness of LonghornLink. For example, just as there is a search feature for campus buildings, a textual search function for events could be included. Furthermore, given our larger purpose of developing a one-stop interface for exploring the UT campus, we might include additional filtering options so that users could see bus routes and parking options (color-coded by permit type), as well as more obscure but useful items such as the location of mailboxes, library bookdrops, etc.

More broadly, LonghornLink could be improved by adding an element of personalization, allowing users to customize the site according to their specific event interests and geospatial context on campus. That, too, might lead to further possibilities of end-user programming, such as user-submitted reviews of campus eateries and other locales on campus. Through it all, the purpose would remain the same: to provide a more thorough, visual and interactive representation of a dynamic campus—and in so doing facilitate knowledge of and likely engagement with the activities and events at the University of Texas at Austin.

6. REFERENCES


