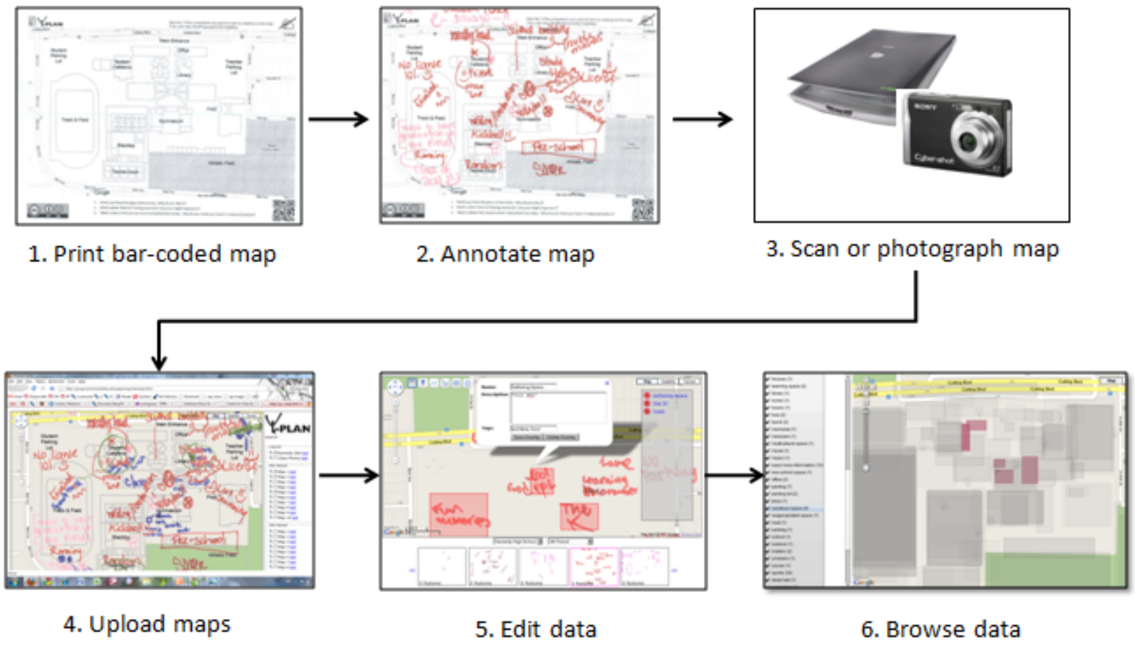
K. Joyce Tsai
Prof. Cecilia Aragon
INFO 247
10 May 2010

The visualization I chose to create was a portion of my final masters project with Sarah Van Wart, Local Ground. The Local Ground system allows users to print bar-coded maps of an area, annotate those maps, then upload them back into the system. After uploading, our code geo-references and image processes the scans and overlays only the annotations on a Google Map. Users can then tag polygons and markers. See Figure 1 for a chart of the system. The information visualization portion of the project involves step 6, browsing the data.

Figure 1: Local Ground system

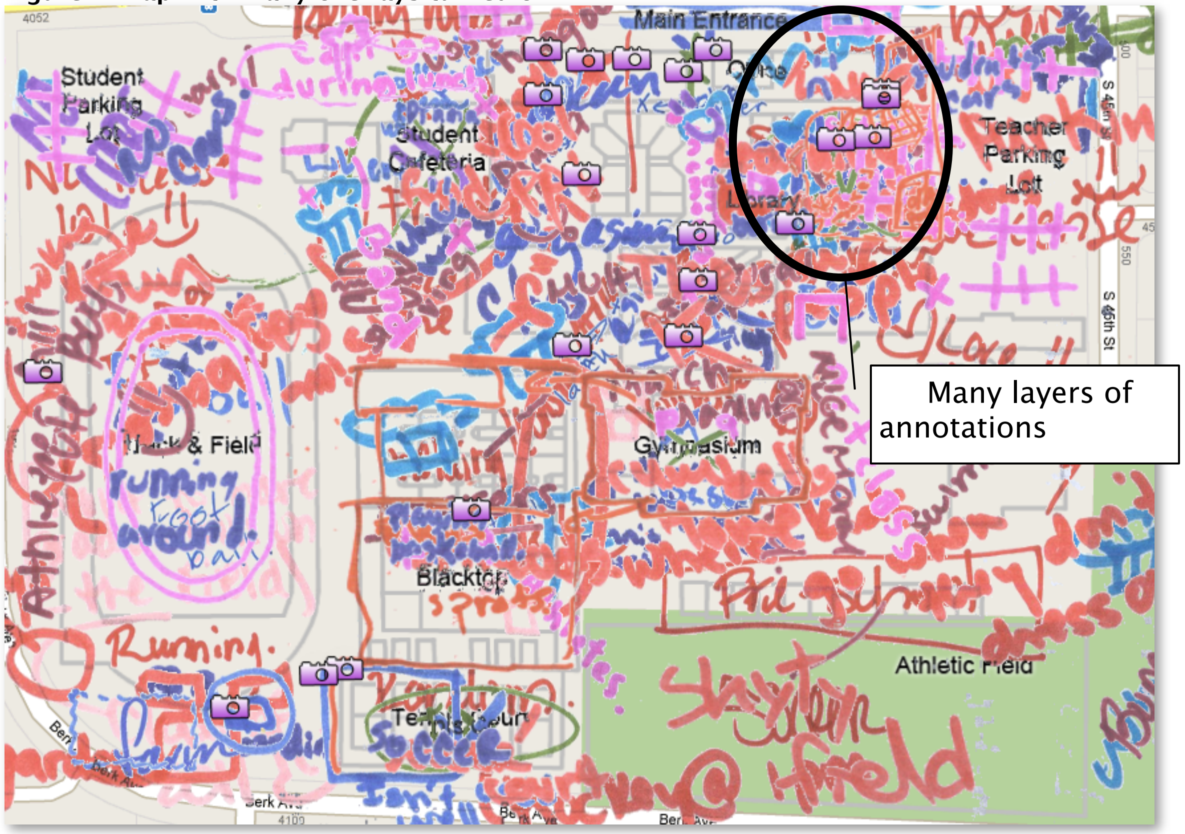


The Problem

After we upload the users' annotations into our system, each user's map is a separate layer. Turning on all the layers makes it very difficult to see patterns in the data, as the individual overlays obscure each other (Figure 2). Although you can find areas on the map that are more heavily annotated than other areas, it is still difficult to tell if comments about a particular area coincide or diverge, or sometimes you cannot even tell what the comments are.

I wanted to come up with a way for city planners to get a summary-level view of the data without losing the ability to drill-down into the data. This way, the intimacy of the hand-written maps would remain even as planners had a higher-level view of the data.

Figure 2: Map with many overlays turned on



Related Work

There are quite a few online map-based visualizations. The ones I looked at most were Ushahidi (http://www.ushahidi.com/) and Oakland Crimespotting (http://oakland.crimespotting.org/), as both are attempts to gather data about local communities in a way that is meant to cast a light upon the living conditions in those communities. However, although both systems rely on information from the ground, particularly Ushahidi with its focus on crises, the information they gather is largely point-based, meaning that it is limited to specific incidences that occur at a particular time and place.

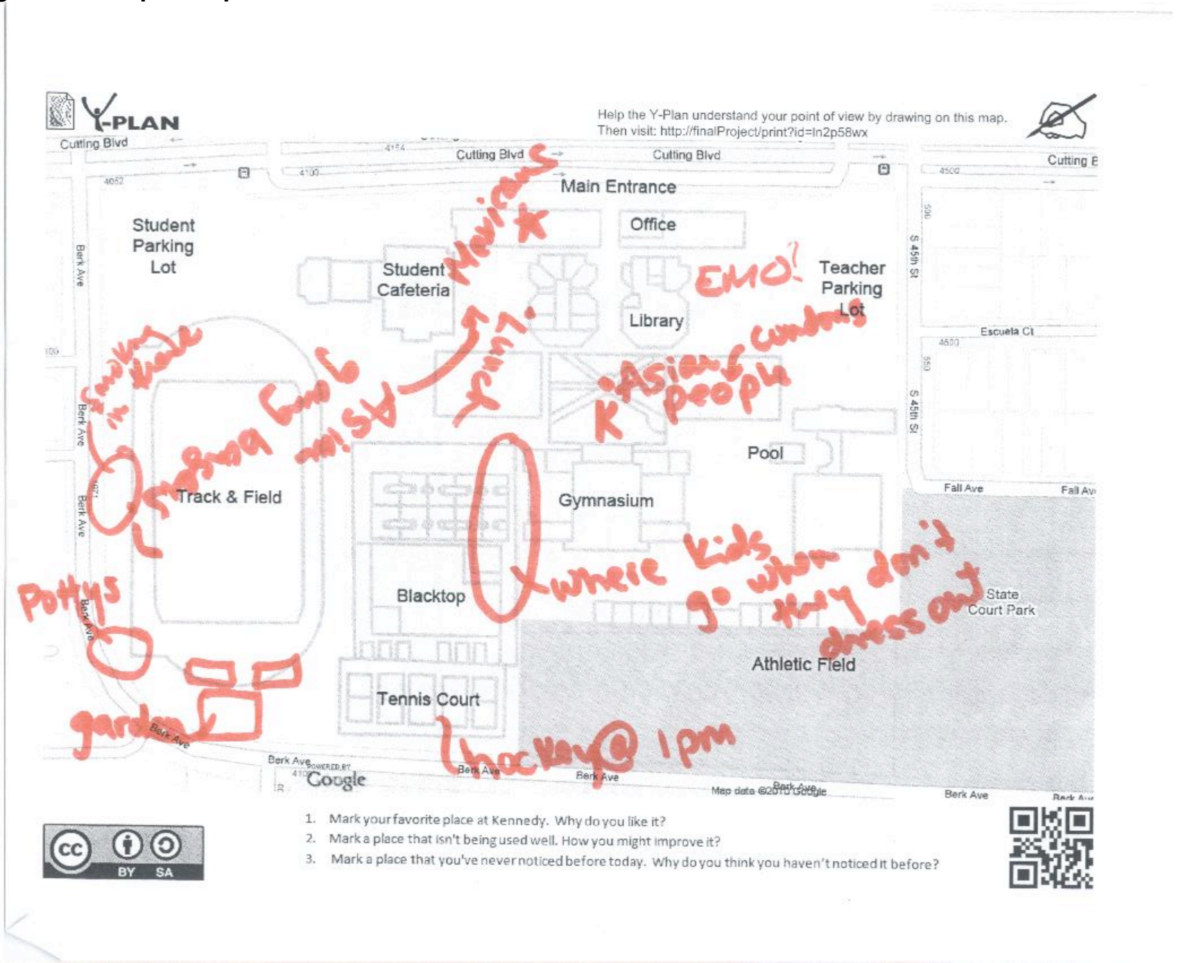
I noted that both systems let the user click on a specific point to get more information about a particular location or incident and that most of the data is organized around broader categories, such as the type of crime for Oakland Crimespotting. While the Oakland Crimespotting user interface allows the user to hover over items in the menu and on the map to see additional items on the map that belong in the same category, many of the Ushahidi applications I checked out were extremely confusing. One particular area of frustration was how the user could continuously zoom in on a point and never seem to get any closer to the underlying data.

Data Sources

All the data in the visualization is from the Local Ground project. Nearly all the mapping exercises were conducted with Kennedy High School and Helms Middle School students, with the exception of the street light mapping in Nystrom Village, which was done by me, Sarah, and two youth council members. Sarah and I also tagged and created polygons or markers for all the information. Currently, the data is being stored on our local server. There is a simple Local Ground API (http://localground.org/api.html) that returns the data in JSON format according to various HTTP Get requests.

In contrast with Ushahidi and Oakland Crimespotting, most of the data we gathered using paper maps is extremely vague in terms of area and in terms of time. For example, in Figure 3, annotations such as "Asian people" seem to refer to a point on the map, whereas the boundaries for "EMO?" or "Asian gang bangers" is much fuzzier.

Figure 3: Sample map


Y-PLAN
Help the Y-Plan understand your point of view by drawing on this map.
Then visit: http://finalProject/print?id=ln2p58wx
Cutting Blvd
Main Entrance
Office
Student Parking Lot
Student Cafeteria
Library
Teacher Parking Lot
Escuela Ct
S 45th St
Pool
Gymnasium
Track & Field
Blacktop
Tennis Court
Athletic Field
State Court Park
Fall Ave
Berk Ave
Mexicans
EMO?
Asian people
Asian gang bangers
smoking in the shade
Pottys
garden
where kids go when they don't dress out
hockey @ 1pm
1. Mark your favorite place at Kennedy. Why do you like it?
2. Mark a place that isn't being used well. How you might improve it?
3. Mark a place that you've never noticed before today. Why do you think you haven't noticed it before?

Because we were simultaneously working on creating the database and the map information visualization at the same time, the initial structure of the data was set by the needs of the information visualization tool. Originally, the only visualization tool we had was one that allowed users to turn different overlays on and off (http://groups.ischool.berkeley.edu/papermaps/kennedy.html). Although this was a good way to demonstrate how the image processing and geo-referencing worked, it did not provide a good summary view of the data.

I first decided to organize the information along the lines of different projects, which would be groups of events centered around a particular location and a particular goal. For example, the NURVE (Nystrom Urban ReVitalization Effort) project was based on renovating a particular neighborhood in the city of Richmond, whereas Kennedy High had to do with mapping events that took place in the high school. It seemed as though city planners would be most interested in project. After the project level, I decided that each project could consist of several different mapping "events," such as street light mapping in Nystrom, or the high school students' field trip to the Martin Luther King Jr. Community Center. Each event would then consist of different scans, which would be the individual map instances. Each scan would then have various associated overlays, which corresponded with different areas marked off on the map. For example, "soccer field" might be an overlay on a scan associated with the 4th period mapping event associated with the Kennedy project.

Tools

I chose Adobe Flex over Javascript and HTML for implementation (I did not have experience in either starting out). The original reasoning was that it seemed easier to build an application from scratch in Flex, since many of the UI components are built into the system. Furthermore, it seems as though Flex had more tools for visualization purposes.

Unfortunately, although Flex may have a more coherent visual library for most applications, the Google Maps API for Flash is much less developed than the API for Javascript. Furthermore, using Flex UI components and repurposing them meant having to overwrite some of the code to try to add functionality, some of which was successful and some of which was not.

The two main downsides of not being a particularly good Flex coder were: 1) not being able to figure out how to incorporate the color-coding within the tag list and 2) not being able to include snippets of the underlying handwritten map in the pop-up information window. Both items were in the initial low-fidelity prototype, and not having them meant the user would not know what polygons and markers a tag was referring to, and, even worse, the user would not be able to see the actual annotation. Because of this, the sense of where the data came from got lost; having a tagger's transcription of the handwritten note pales in comparison with being able to see the handwritten note and illustration on the map itself.

Low-Fidelity Prototype

I created the first prototype using Balsamiq and PowerPoint after briefly investigating the different UI components available in Flex (Figure 4). The first prototype organized the tags using a checkbox tree, where the user could expand and hide the different overlays under each tag. The color assigned to each tag would be displayed next to it, and clicking the checkbox for the tag would check all the corresponding overlays underneath. When an overlay was turned on, the user would be able to click it to bring up an information window containing a transcribed description of the annotations, along with a picture of the annotation (Figure 4).

Hovering over a tag in the checkbox tree would gray out any overlays not associated with that tag, while hovering over the overlay name in the checkbox tree or the overlay itself would cause all other overlays to gray out (Figure 5). I also had the idea for a summary view that appear in a lightbox over the map (Figure 6). The summary view would include the underlying annotations, descriptions, and tags for each item in the list that was checked.

Figure 4: Balsamiq prototype

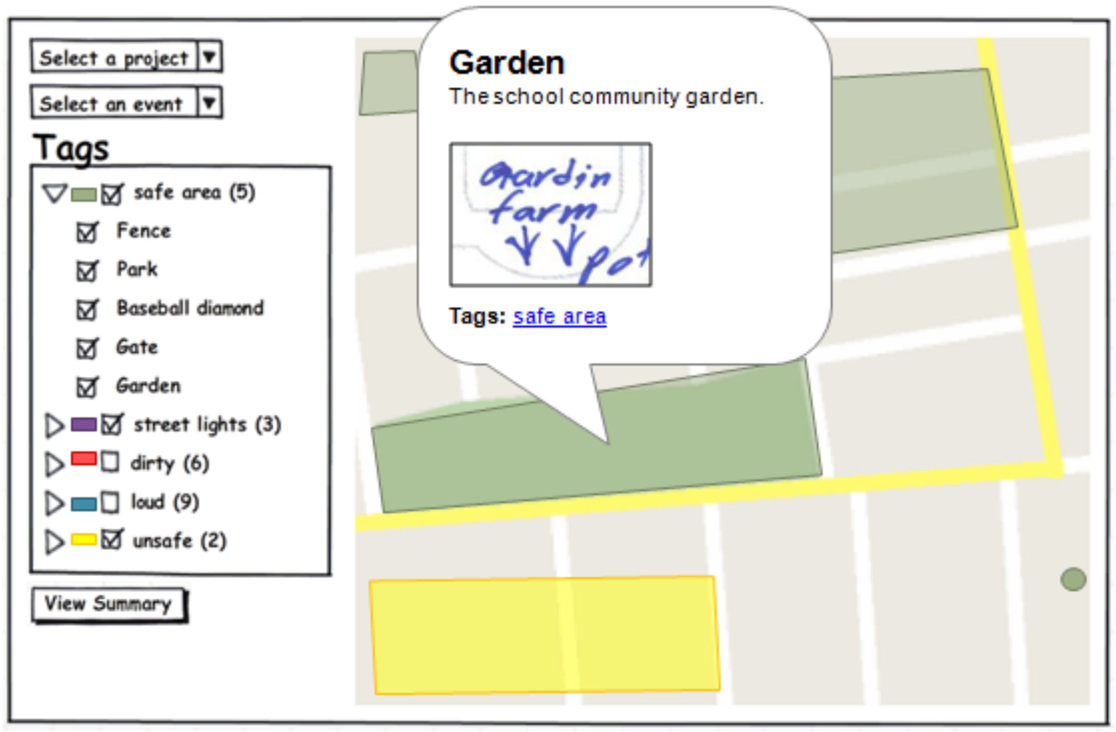


Figure 5: Balsamiq mockup on hover

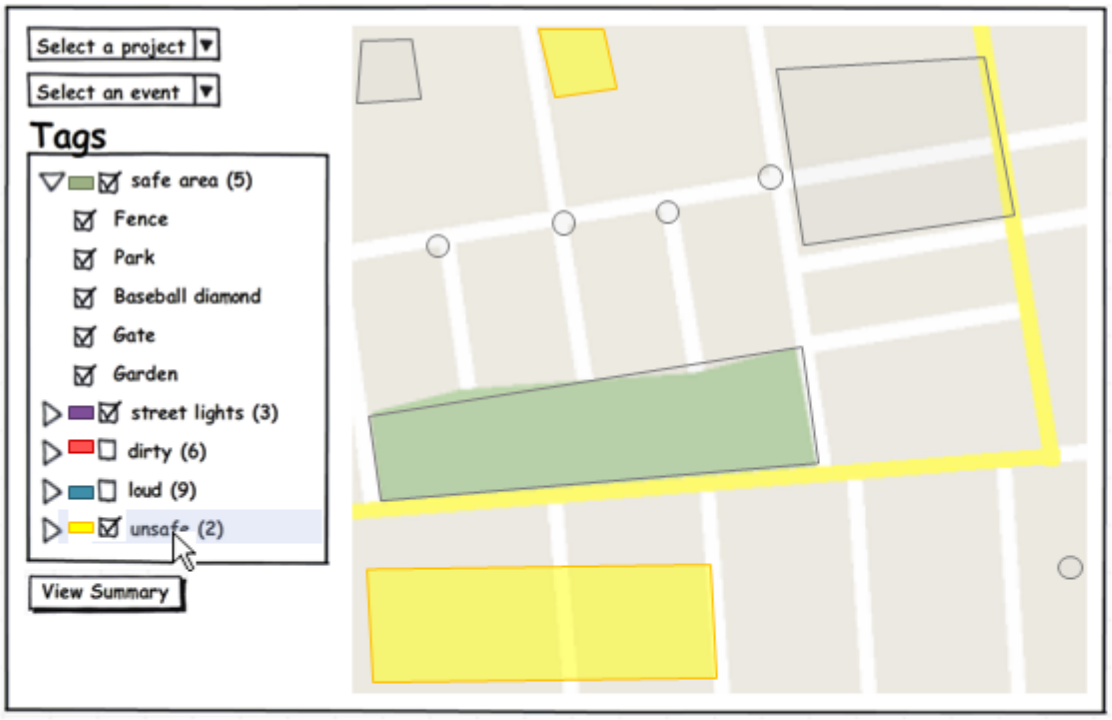
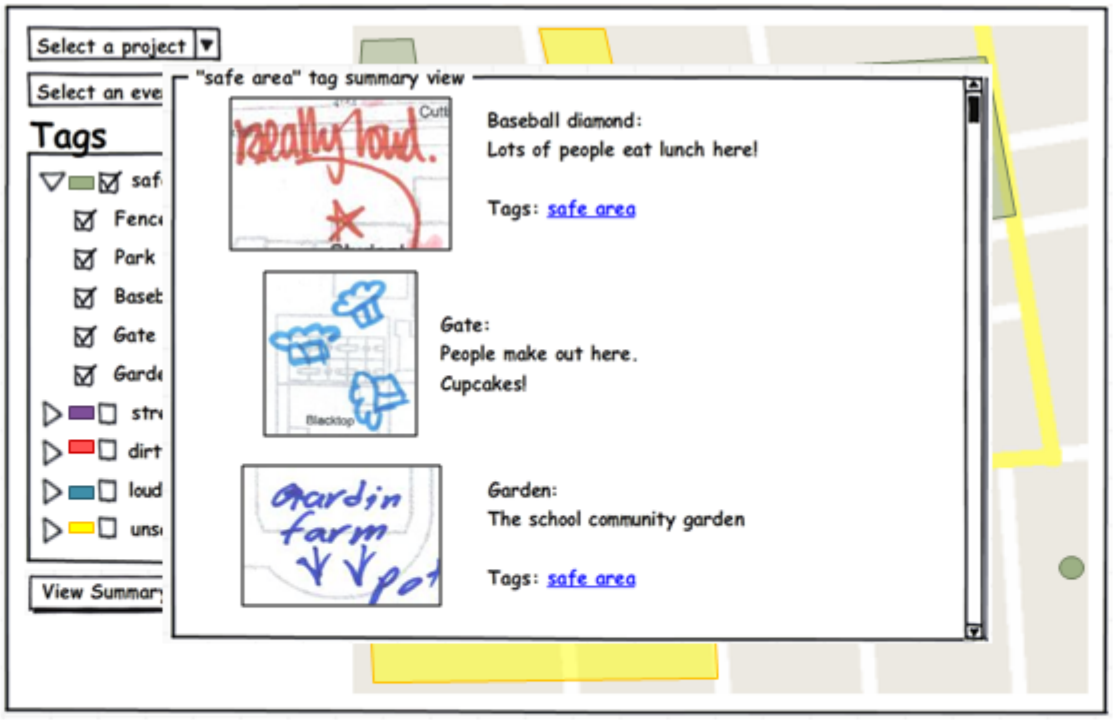


Figure 6: Balsamiq prototype summary view



Baseball diamond:
Lots of people eat lunch here!

Tags: safe area

Gate:
People make out here.
Cupcakes!

Garden:
The school community garden

Tags: safe area

User Testing, Round 1

I tested the low-fidelity prototype on an employee of the Richmond Children's Foundation, an organization involved in the NURVE program. As such, she frequently viewed plans from the city of Richmond regarding the renovation of the community center and park. She had previously noted that in the past year, the city had organized a community mapping exercise asking community members to map out their ideal park. The employee then had to take the resulting ten paper maps and somehow summarize all the information in the Word document, something she said was extremely difficult. Her methodology for summing up the information was basically glancing from one paper map to the next and trying to keep the information in her head.

When I showed her the low-fidelity prototype, she understood the tag list and how checking and unchecking the boxes would turn overlays on and off. She also noted that the hover effects and the tool tips that appeared with a mouse rollover seemed intuitive. However, she stated that she also wanted more about who came up with the annotations and who had tagged the map, as she would weight the opinions of community members and official planners differently. She also said that although the summary view was very useful, she did not want to it obscure the entire map. She noted that it would be extremely useful to have the summary view to print out and bring to meetings in order to illustrate a point.

Flex Prototype

The prototype in Flex ended up incorporating less of the RCF employee's user feedback than I would have liked. Although my final project partner and I had both thought about incorporating the identities of whoever had created the maps in our database, when we gathered the data, we did not ask the students to put their names on the maps for fear of privacy issues. Some students had marked where their lockers were, and in the mapping exercise of the Nystrom neighborhood, some students had marked where their house or their parents' houses were.

With the RCF employee's feedback, I had planned on changing the format of the summary view from a pop-up lightbox to a sliding panel that could be hidden away if the user wanted a larger view of the map. However, because I could not manage to get the underlying map annotations to show up in Flex, the summary view was never coded.

Also, while coding the Flex prototype, I discovered that by using tags instead of more set categories, we had the problem of one overlay being associated with several tags. In this case, if the basketball court were tagged "sports" and "cute guys," checking both tags would create two identical overlays. If the user wanted to turn off that overlay, he or she would have to turn off both "sports" and "cute guys." I thought about using only single overlay despite overlapping tags so that turning off that overlay would turn it off for all other corresponding tags (i.e. turning off the basketball court for "cute guys" would also turn it off under "sports," even though the user never clicks on "sports"), but that behavior seemed counter-intuitive. After consulting with Ljuba over several ways to display multiple tags associated with a single overlay, I decided to stay with the current method of simply having several instances of the same overlay turn on if it was associated with various tags. That way, the opacity of the overlay would convey that it had more tags than an area that was less opaque and layered.

I also changed the interface from a tree-based menu to a menu with flyouts. When I had first designed the interface, we only had sample data in the database. However, once we ended up tagging all the scans, we found that the list of tags was extremely long. As such, using the checkbox tree would only make an already long list even longer, making it even more difficult for the user to find a tag they wanted.

User Testing, Round 2

The second round of user testing used a partially complete version of the Flex prototype. I tested it on two members of the program working with the high school students. They were familiar with the geo-spatial data we had collected and were working with the students to create a high-level overview of the mapped areas to present to the Richmond City Hall. However, because I tested without real data, some of the reactions were a little difficult to gauge. The list of tags had been made up, so they were not as relevant to the users. This is also where the lack of the underlying annotations showed. Both users were much more excited when I showed them some of our data from the NURVE visioning event, where they could see photographs that the students took. Although clicking on the polygons to get more information was nice, being able to see more media embedded in the map was the true draw. The hover action and the checkboxes for the tags seemed to make sense for both users, as did the use of polygons and markers.

Future Work and Conclusions

As you can tell, there is still a great deal of work to be done on the prototype. Due to the limitations of Flex, I would like for any future work on the viewer to be done in Javascript. Because our system currently does not have thumbnails of the annotations corresponding with different polygons, the only way to display the underlying maps so far is to embed a smaller Google Map within the information window. I could not manage to get that working in Flex, which greatly weakened the power of the visualization tool. Similarly, the lack of a color legend for the tags goes against the basic principles of information visualization.

Still, it seems as though the basic principle of having translucent colored overlays made sense to all the users, despite most online mapping services relying solely on points. The way the overlays overlapped to show the relative density or paucity of information in an area also seemed to be fairly intuitive, as did using a mouseover to discover more information about a particular layer. There are also some issues regarding tagging and controlled vocabulary that are beyond the scope of the map visualizer but nevertheless impact the visualization. Although our current tagger (http://localground.org/map-editor) automatically suggests tags based on what tags have already been used in a system, there is no controlled vocabulary in the background to tie together tags such as “gathering space” or “hanging out.” As such, people using the visualizer must sort through synonyms themselves or toggle layers on and off to see if there are any correlations between similar tags.