Dishmaker: Personal Fabrication Interface

Leonardo Bonanni, Sam Sarcia, Subodh Paudel, Ted Selker
MIT Media Laboratory
20 Ames Street
Cambridge, MA 02139
+617 253 4564
amerigo@media.mit.edu, ssarcia@mit.edu, paudel@mit.edu, selker@media.mit.edu

ABSTRACT

People are tool-makers, but today we often depend on centralized fabrication for our tools. This paper explores the user experience opportunities demonstrated in a case study of a computer-controlled domestic fabrication system. Specifically we explore several approaches to creating a dish-maker that would create dishes on demand under computer control. The Dishmaker can create cups, bowls and plates and recycle them into their raw material when the user is finished eating. A graphic interface allows users to select between cups, bowls or plates that can be created in any volume. This paper describes several working prototypes and shows the approaches for defining your dishes for a meal.

Author Keywords

Tangible computing, Personal Fabrication, Machine Tool, Appliance, Product Design.

ACM Classification Keywords

J.7: Consumer Products.

INTRODUCTION

Rapid prototyping techniques can produce an increasing variety of forms and materials based on 3-dimensional computer models. Personal fabrication could one day produce everything we need locally, replacing the transportation of atoms with the digital transfer of designs. Current personal fabrication are expensive, slow and take considerable attention to use. For making these useful for daily use they should have simple physical and computer interfaces, require less effort than not using them Furthermore, there is little market for *everything* to be manufactured locally, in fact, many of the objects we buy and use have a long useful life. Yet our homes are cluttered with staples that we collect in case they one day serve their purpose. Dishes in the kitchen are one example of an object that actually wastes energy by having a long product life. Aside from the cost of production of infinitely durable plates and bowls, dishes require frequent washing for the duration of their lives – not to mention storage with its associated materials and space. By targeting this specific problem in the kitchen, we are seeking to produce a personal fabrication interface capable reducing the amount of things we live with. The scope of the idea is to obtain new dishes on demand for eating and to be able to recycle them back into the system. The term "dishmaker" was chosen to reflect the potential to replace a large part of what a dishwasher does with a more fundamental recycling effort. The mechanism is called a dishmaker because it produces the useful plates, bowls and cups that can be used for eating. It recycles them so that they can be re-produced for the next meal. By storing the dishes in their raw material, the dishmaker seeks to eliminate clutter as well as to replace storage spaces with productive space.

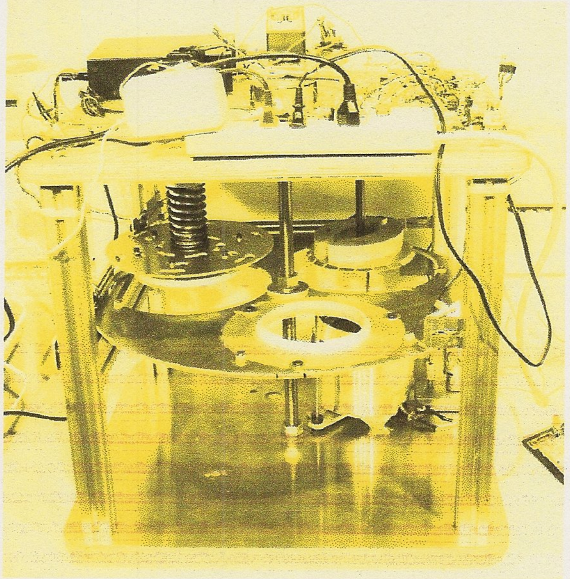


Figure 1. Dish Maker

CHI 2005, April 2-7, 2005, Portland, Oregon, USA.
ACM 1-59593-002-7/05/0004.

RELATED WORK

Today many prototyping laboratories include expensive stereo-lithography, inkjet-stabilization model creation systems and fuse-deposition modeling systems that slowly produce highly detailed models from computer instructions. The idea of rapid prototyping has led to a number of different products already affordable by research laboratories and educational institutions. These employ sophisticated modeling and fabrication to produce accurate, complex forms in small quantities over large periods of time. At the same time, the objects are homogenous lumps of paper, plaster or plastic and require intensive training and post-processing. These objects cannot be recycled locally in the same way that they are produced.

In recent years, the concept of "cradle-to-cradle" manufacturing has been proposed by McDonough and Braungart to describe production design that consider the entire life-cycle of a product. One of the lessons of this theory is that "trash=food:" the end of a product's life should be designed to serve as the beginning of another. This is even more useful if it can be achieved locally, for example if a biodegradable soda bottle contains a flower seed and will actually beautify the landscape when thrown out of an automobile window.

By combining the idea of personal fabrication with the "trash=food" philosophy, we can begin to design appliances that produce consumer goods locally while designed to recycle them into new, useful products.

DESIGN

When discussing dream kitchen projects we expressed the desire to throw away dirty dishes after each meal and get new ones for the next. We interpreted her wish as a micro-factory capable of variable production and recycling within the envelope of a conventional dishwasher. Theoretically, such a device could replace part of the cabinets and dishwasher since the material would return to storage in its raw form. The question was whether we could make a variable molding machine capable of producing multiple objects rapidly and recycle them for re-production. Unlike a rapid prototyping machine, this micro-factory would have limited variability but unlimited speed and volume. In addition, it would have to recycle the products indefinitely.

Fortunately, dishes are not as variable in form as typical rapid prototypes. The geometric family of concave, waterproof containers can be produced in a number of ways. Initially, we considered a pottery wheel-approach whereby a robotic arm could throw clay on a robotic pottery wheel. A polymer-clay composite was researched for this purpose, but the need to melt and re-process molten plastic complicated the device into a small injection-molder with high energy consumption. Next we investigated the potential to blow-mold the material, but this alternative required a compressor. Finally, we constructed a machine the size of a dishwasher that can stamp, dispense and recycle plastic pucks into dishes of various sizes with simple components already found in most kitchens: electric motors, heating coils, and micro-switches. Multiple prototypes and proofs-of-concept were involved in the design by a multi-disciplinary team of six mechanical and electrical engineers. A simple interface was developed to "order up" dishes for a meal. Future work will seek to increase the variability and quality of dishes while reducing the energy and time required to make them. Ideally, the dishmaker will become an all-in-one cabinet that contains anything you need, produced the moment you reach in.

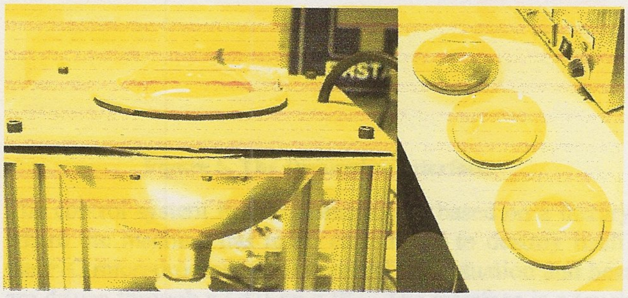


Figure 2. Dishmaker 1.0 showing variable pressure-mold (left) and various dishes (right).

Prototypes

First we demonstrated that a round disk of acrylic could be formed into a dish, plate bowl or cup. We did this by clamping an acrylic disk in a ring, heating it with a lamp and pressurizing it to blow a dish, bowl or cup in a pressure-forming system (see Figure 2). We sought a non-toxic material with a low softening point, and ended up with acrylic wafers. Acrylic and other amorphous polymers have "shape memory," meaning that under certain conditions the material returns to its original form. This property eliminates the need to melt the plastic, reducing the energy consumption dramatically. We proved the wafers could be recycled many times in a sandwich press under slight heat and pressure (see Figure 3). Our next goal was to make a system that would be computer-controllable allowing a dish-making interface

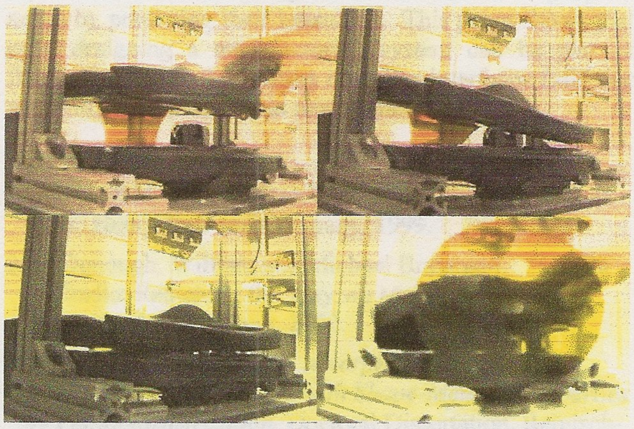


Figure 3. Recycling dishes in a sandwich press.

PLACEMAP:
A USER-BASED MAPPING APPLICATION

by

Jeffrey D. Hoff

Under the supervision of

Professor Ted Selker

Media Laboratory

Massachusetts Institute of Technology

May 23, 2006

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

ABSTRACT

placeMap: A User-based Mapping Application

by Jeffrey D. Hoff

The placeMap project is a user-based mapping application for easily viewing people, places, and events. This paper describes a system designed and implemented at the Massachusetts Institute of Technology, but expandable to any organization. By using the Adobe Flash version of the Yahoo! Maps API, the project realized goals of being a cross-platform, easily distributed solution for geo-locating events and people. The application allows users to instantly communicate with friends and view events and people at their respective locations. This project shows the possibility of creating user-based maps that are customizable for each user's preferences and goals without a major computational overhead.

i. Table of Contents

ii. Table of Figures

1. Introduction

1.1 Motivation

Cartography has been an extremely useful tool for over 4300 years. The Babylonians created the earliest known maps around 2300 B.C. on clay tablets. The maps of today contain much in common with those of the Babylonian time period; most are flat depictions of an irregular surface that retain focus on the place that they are portraying. However, technology now allows us to change the map dynamically, customizing the map to each user that interacts with it. Before one can utilize this ability, he must have an understanding of place and time, with a strong knowledge of how they affect a person. This need to comprehend the significance of place and time as they influence people forms the basis of this project.

The original reason for the placeMap project was to merge the ideas of maps and user based systems together. The hope was to better understand how people view time and place, and how this in turn affects them. CampusMap is the actual project that I undertook and aims to create an application to realize the goals of the placeMap project.

1.2 Goals

The CampusMap project has several main goals. The principal objective is to create a system that allows users to visualize events, people, and location together. Because this is going to be a research tool as well as a usable application, there must be an easy interface to allow for expansion. There must be several channels that allow effortless data collection, as this information will be the main criteria for future studies. As for the application itself, it should be generic so that other schools or groups may employ it in their own location. To complement the option for others to implement the software, there should be expandability to allow for the display of other types of data.

In terms of usability, the application should allow customization. Because this is a user-based system, the user should have control over the data that is displayed such as filtering, sorting, and placement. Security of information is of low priority since the service will keep no crucial personal information, eliminating the need for high-encryption processes such as Kerberos.

1.3 Previous work

The placeMap project was by no means new when I joined. There have been multiple generations of the project that were created and tested, yet never released. Initially started by Ben Piper at the MIT Media Lab, the project has matured in the Context Aware Computing lab. I, personally, was involved only with the two most recent generations of the project.

2. Design

2.1 Overview

The premise of the architecture was to build a user interaction layer on top of the Yahoo! Maps API. I was encouraged to use the Yahoo! API over other possible APIs because of the Adobe Flash[1] platform. The Flash Player[2] is cross platform and ideal for web-based deployment solutions, ensuring that virtually anyone with an internet connection can view the application with the free software. The platform holds one

[1] Adobe (formerly Macromedia) Flash® is a development tool that was used to create the animated online website. Macromedia Flash is produced and trademarked by Adobe Systems Incorporated [http://www.adobe.com].

[2] Adobe (formerly Macromedia) Flash Player® is the free client to view Flash files.

major advantage above competing open source mapping APIs that are typically produced in plain JavaScript[3] or Ajax[4]. Flash uses vector graphics, promotes animation, and compresses file sizes. Vector graphics greatly reduce the file size and ease of producing high quality images that do not distort when resized. Its smaller fast transmission speeds and on-the-fly rendering simplify animation procedures by not forcing the developer to reduce user interaction for the sake of size. Using the provided functionality of the API also facilitates the ability for the user to choose a satellite, standard map, or hybrid view of that location, further customizing the user's experience.

These factors all aided in the decision to use the Yahoo! Maps API, which implements the Flash platform; however, the core reasons are guided by the goals for the project. First, the Flash-enabled API allows cross-platform accessibility without any additional effort. Next, using Yahoo!'s map allows the application to literally post objects at any location in the world without limit. This ability ensures that this part of the system is portable to any other organization that wishes to use it. Lastly, anyone can download the software needed to run our application easily and free of charge, including as many potential users as possible.

2.2 Preliminary Design

The preliminary design was soon incompatible with the implemented system as a whole. Because new features were constantly being added to the project, the initial design became more outdated as the project matured. Classes and objects began to

[3] JavaScript™ is a scripting language used mostly in websites to aid in user event handling and is a registered trademark of Sun Microsystems, Inc.

[4] Ajax, short for Asynchronous JavaScript and XML (though its creator Jesse James defends that is not an acronym), is a technique used to use asynchronous calls to a server to reload only small parts instead of the webpage as a whole, saving both time and transfer traffic. Through this Ajax enhances the user experience by making data transition appear seamless.

contain circular-dependencies causing both troubleshooting and project understanding to become nearly impossible. The project had no version control system in place, which made reverting to previous versions nearly impossible, except for saved copies in .zip files. These two major reasons, among others, brought the development of placeMap to a necessary decision point.

The two options were to try and fix the current version, or to completely redesign and re-implement everything done so far. The easy answer was to not throw all work away. However, the smart choice was to start over with an intelligent design using the knowledge gained from the first attempt. Before jumping into the creation of a new design, several users participated in an ad hoc questioning session about the usability and feel of the user interface. Using the information from the first effort and the input from users, the new design was structured better and visually more appealing.

2.3 Reimplementation – A New Design

The new, current architecture of the system still lies on the Yahoo! Maps API, enabling placeMap to be lightweight and visually attractive. The placeMap project as a whole can be categorized into three distinct components: client software application, Jabber[5] server, and web services. The client software application is by far the largest component, as it contains all of the tools for user interaction with the system. The Jabber server handles the chat exchange between users, providing some security in that user clients do not talk directly to one another. The web services are a set of ASP.NET tools that allow easy reference to data manipulation and extraction methods.

[5] Jabber® is an open source, server-based method of instant text-messaging using XML-based communication.

The class structure intelligently enforces a lack of circular dependencies and a central data-based architecture. Figure 1 below depicts the modular dependency diagram in which you can see the consistent hierarchy. Figure 2 shows the various Data Brokers that the Yahoo! Map depends on for information to display. Each of the three brokers is responsible for a specialized type of data storage and recollection, namely event, person, and location. The individual classes will be described later in section 3.

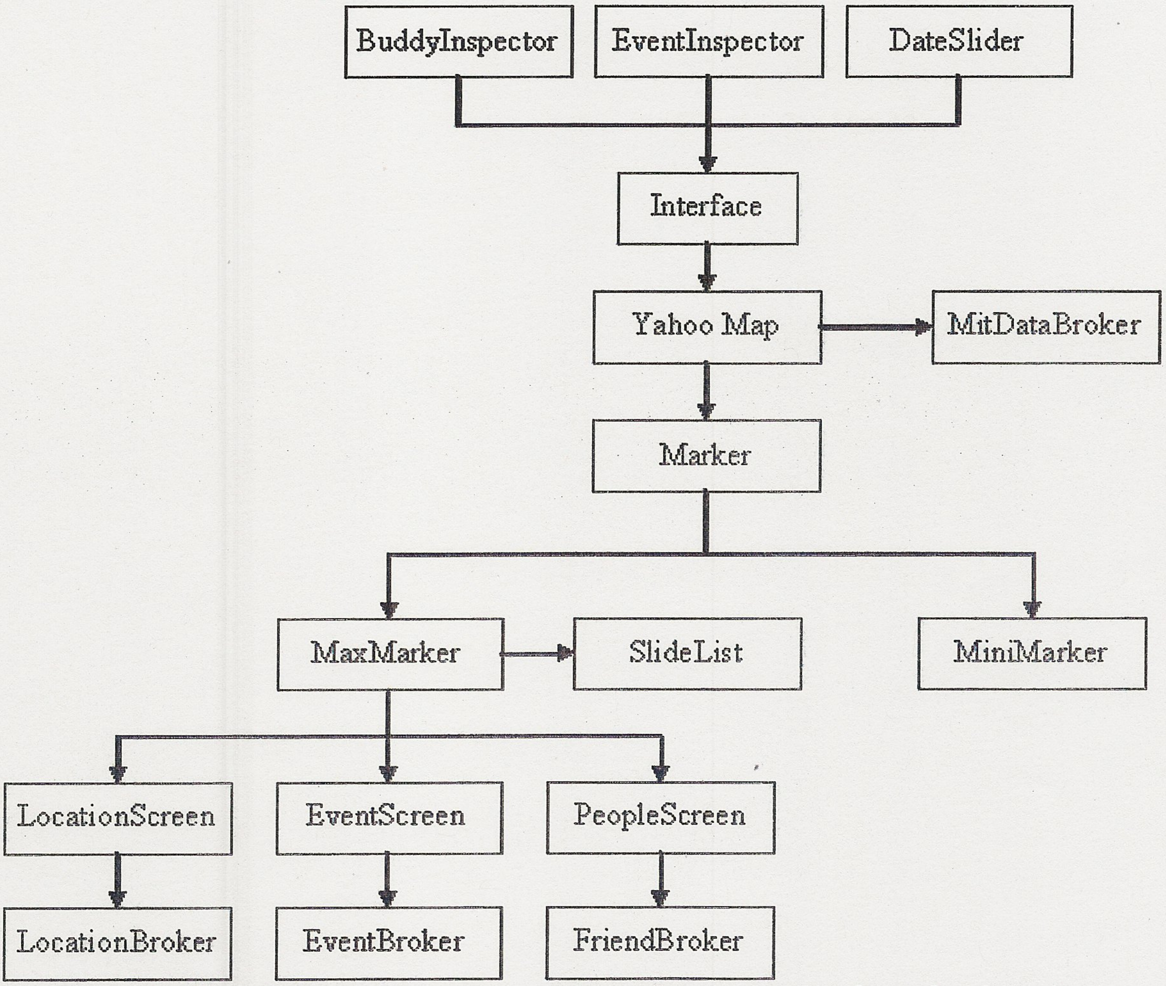


Figure 1. Module dependency diagram. Note that no circular dependencies exist

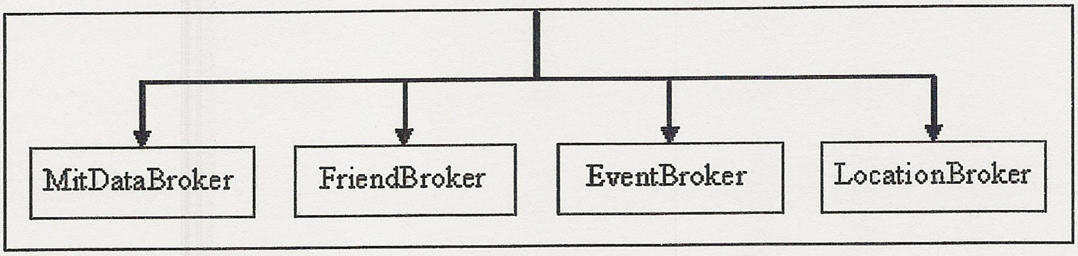


Figure 2. Diagram of the Data Brokers.

3. Architecture

3.1 Client Software Application

The client software application was programmed in ActionScript[6] class files, as opposed to within the movie time frames. The main extension that placeMap adds to Yahoo!'s Map API is the novel marker described below.

3.1.1 Marker

The first figure, Figure 3, shows a screenshot of an unexpanded marker. The program displays this as an "icon" to inform the user of underlying information. The program tinted one of the markers maroon to indicate the user's current geographical location. Simply rolling over the marker with the mouse will cause the marker to animate into the expanded state. Figure 4 shows a brief glimpse of the marker expanding from small to enlarged views.

[6] ActionScript is the scripting language used to program Flash. Its roots are based on the same standard as JavaScript, assisting in fluency in the other if one is known.

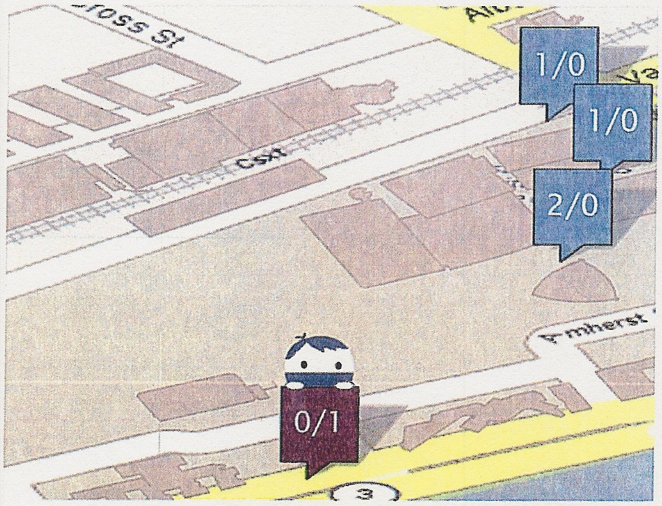


Figure 3. Unexpanded marker

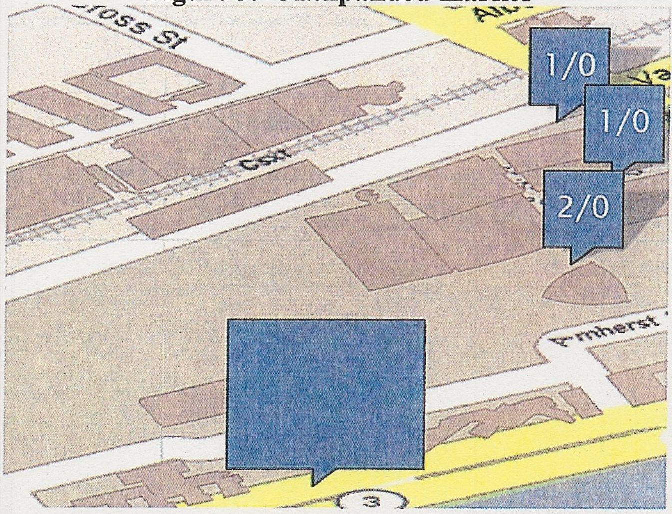


Figure 4. Marker in transitional state during animation

Once the animation is complete, the expanded marker is visible to the user and depicts information relevant to the selected location. In the example of Figure 5, a user is at the location in question. Since this example shows the current user, only basic information is provided. In the case that a person other than the user is selected, a chat window overrides the information screen and allows the user to chat with the other person via the Jabber instant messaging system. When the marker does not have the focus of the mouse, the small marker "throbs" to alert the user about an incoming message. This throb effect is achieved by animating the marker to become bigger and smaller repetitively until the user checks the message. By using this visually pulsating effect, the location not only catches the user's attention, but actually appears to be reaching out to the user. In this respect, instant messaging is no longer communication with a person at an unknown place, but a conversation between places.

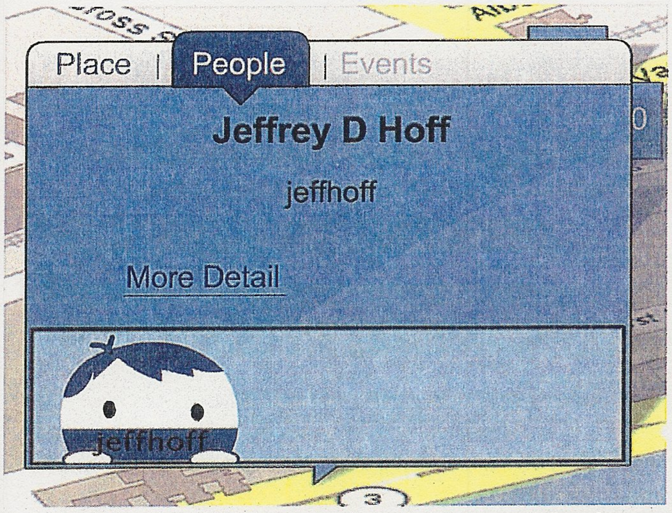


Figure 5. Marker in expanded state with people view open

When a person is not at the location in question, the events screen is shown, which displays the details of an event. In Figure 6, a marker is expanded showing a prayer and bible study at the student center. The image displayed as its icon is a randomly generated image from the web, acquired from Yahoo! Image Search.[7] Through the service, images matching event keywords are returned and a random picture is selected as the icon for the event.

[7] Yahoo! Image Search is a web search utility to find images on the web with certain descriptive keywords. For more information or to use the free service, please visit [http://search.yahoo.com/images].

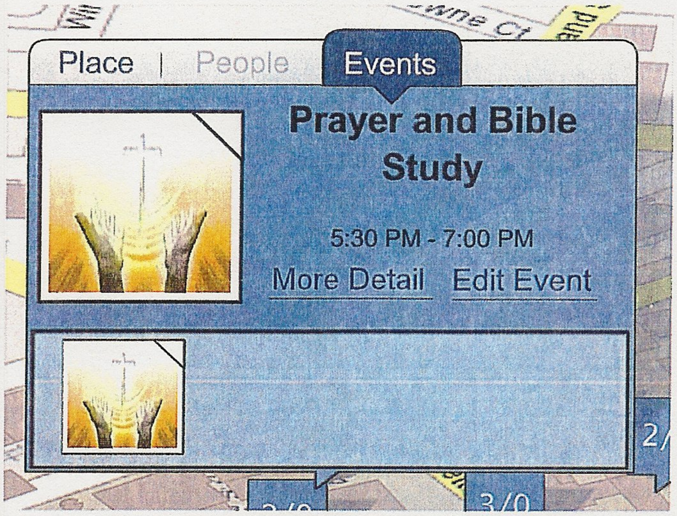


Figure 6. Marker in expanded state with event view open

3.1.2 DataBrokers

In this project, a data broker is defined as an object that's sole responsibility is to collect, store, format, and distribute data between a web service and the internal application. placeMap has four data brokers – FriendBroker, LocationBroker, EventBroker, and MitDataBroker. The MitDataBroker is a container class to hold the data brokers, while the other three perform all tasks related to retrieving information from the centralized server, and storing it internally to the application for efficient use.

3.1.2.a FriendBroker

The FriendBroker class is responsible for handling all of the user information, including friends, current user, and potential friends. In this respect, it is dependent on the Person class, and inherently on the Location class. The FriendBroker allows for such functionalities as getting a list of friends, adding/removing friends, and updating user information. One of the most important functions of the FriendBroker is the ability to determine both location of the user, as well as the locations of his friends. The correct placement of all person markers is dependant on this information.

3.1.2.b LocationBroker

The LocationBroker handles all information regarding locations stored in the database memory. This is heavily dependent on the Location class that stores all information about a given Location. The LocationBroker is highly needed for the methods to retrieve all buildings on the map, and reverse lookup of buildings based on building number. An example of this would be placing a friend on the map at the proper location, after just receiving their data.

3.1.2.c EventBroker

Possibly the most important of the data brokers, the EventBroker communicates with the database to retrieve and process the events from the database. EventBroker handles all events as it retrieves, sorts by location, and displays the events on the map. This forms the basic functionality of placeMap.

3.1.3 User Interface

The user interface has three main components other than the actual Yahoo! Map. The first is the DateSlider that enables the user to view events that occur on different days. Next is BuddyInspector which acts as a "buddy list" that shows current users and the online/offline status as well as allows a user to add or remove friends. Last is the EventInspector which is the most complicated, yet most informative of the group.

3.1.3.a DateSlider

In order to view days other than the current one, the DateSlider provides the functionality to select a desired day to view. Its default and initial position is “closed” in the bottom right hand corner of the screen, simply showing the date that is currently being viewed. If a user clicks on the DateSlider, it slides to the left, expanding to show six days forward and backward from the current date. Clicking a desired date will slide the DateSlider back to its compact state in the corner and refresh the view on the map to display the events on the newly selected date. Figure 7 shows an example of the DateSlider in expanded view with Thursday the 18th selected.

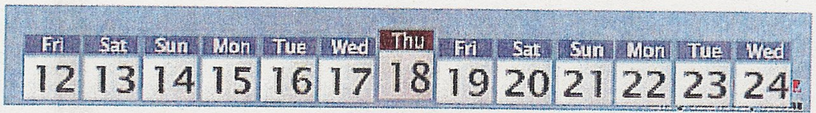
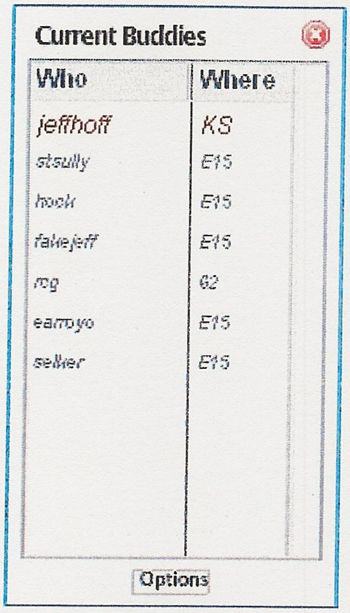


Figure 7. Screenshot of the DateSlider when expanded

3.1.3.b BuddyInspector

As is typical in a program that allows instant messaging, a user needs to know the online/offline status of his friends and how to reach them. The BuddyInspector handles this need by displaying the current user and his friends along with their locations. The marker at the location of the current user is tinted maroon to avoid confusion between friends. When a user is offline, they are smaller, faded out, and italicized, unlike the online state that is black, non-italicized and of bigger font. Clicking on a user centers the map on their location so the user can easily find them or talk to them, as well as opening the Billboard to display all of the known information about the user. The BuddyInspector also has an options menu that allows the user to add or remove friends from his buddy list.



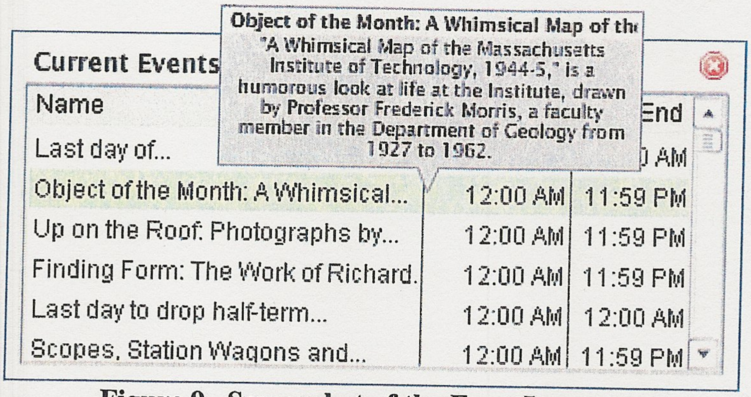
Current Buddies

Who	Where
jeffhoff	KS
stsully	E15
hook	E15
fakejeff	E15
rog	62
earroyo	E15
selker	E15

Figure 8. Screenshot of the BuddyInspector

3.1.3.c EventInspector

When acquiring events from online sources, not all have a mapped location on the system. This means that not all events are visible as markers on the map. To assure that a user can find any event in the system, the EventInspector was added to the map. As is shown in Figure 9, the EventInspector concisely presents the title of the event along with the start and end times. Because ease of use is of utmost priority, a tooltip-like explanation window follows the cursor and contains the description of the event the mouse is hovering over. Clicking on an event will have a two-fold effect similar to that of the BuddyInspector. First, the map will center on the location of that event if the system knows it. Secondly, the Billboard will display all of the known information about the event.



Current Events

Object of the Month: A Whimsical Map of th

"A Whimsical Map of the Massachusetts Institute of Technology, 1944-5," is a humorous look at life at the Institute, drawn by Professor Frederick Morris, a faculty member in the Department of Geology from 1927 to 1962.

Name		End
Last day of...		0 AM
Object of the Month: A Whimsical...	12:00 AM	11:59 PM
Up on the Roof: Photographs by...	12:00 AM	11:59 PM
Finding Form: The Work of Richard.	12:00 AM	11:59 PM
Last day to drop half-term...	12:00 AM	12:00 AM
Scopes, Station Wagons and...	12:00 AM	11:59 PM

Figure 9. Screenshot of the EventInspector

3.2 Jabber Chat System

Jabber is an open source, server-based protocol of instant messaging between users. Its communication is XML-based, which allows it to be easily extended to handle customized message types. The key factor in deciding to use a Jabber server as the means for inter-user chat was the ease of creating a private, or public, network for users to join. This capability is one of the highlights of the Jabber project, allowing anyone to setup a server, create users, and control who can communicate and how. The open ended possibility of linking networks together is even more exciting, as this could become a future goal of the placeMap project.

Based on the Jabber protocol, XIFF is an open source chat client written in Flash ActionScript. Because it is open source, its code was modified to ease the integration of XIFF into the placeMap architecture. The chat-handling module distributed with XIFF was too bulky and bogged down with many unnecessary features. Additionally, it lacked certain features integral to the system. For this reason, an entire new chat-handler was built, which controls all communication between placeMap and the Jabber server.

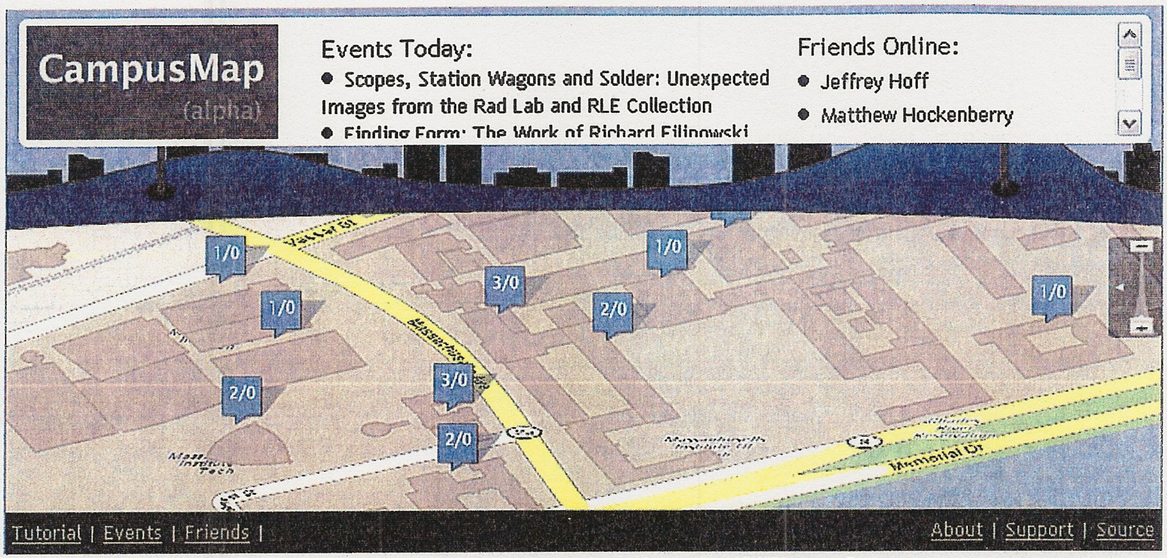
3.3 Web Services

3.3.1.a SOAP-based web services

The web services of the placeMap project are an integral part of the system as a whole. They not only create an easy method for centralizing user information, but abstract away the database completely to the client application. From the viewpoint of the client application, the web services are merely another method that is available to call. Written in C# .NET, the web services are an extremely fast communication link to extract database information. To interface with Flash, the web services use XML-based SOAP calls to correspond between computers. Although typically slower than binary protocols, the SOAP messages provide an extremely easy method for transfer of small data messages.

3.3.1.b Billboard – website driven information

One of the most considered features of placeMap is its look and feel. One goal in this project was to make the user as comfortable in the application environment as they would be in a normal setting. Several visual modifications assist in this; however, the Billboard is the most useful of them. Figure 10 shows the opening screen of the Billboard, which alerts the user to some random events of the day and shows his friends that are currently logged in. A more prominent purpose of the Billboard is to "advertise" the information about a selected person, place, or event. As previously stated, when a user or event is selected in their respective inspector, a webpage displaying all of their information is expanded in the Billboard. Figure 11 displays an example screenshot of the previously shown event.



CampusMap (alpha)

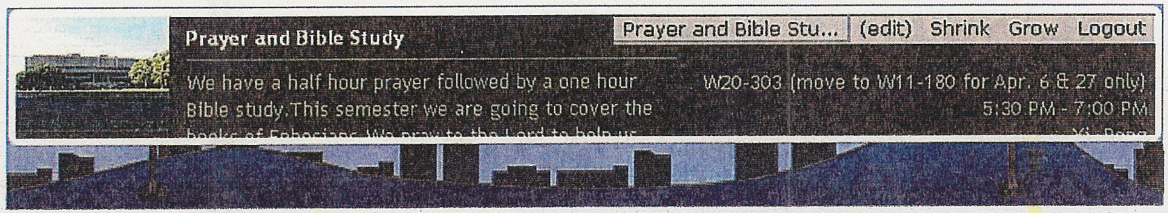
Events Today:

- Scopes, Station Wagons and Solder: Unexpected Images from the Rad Lab and RLE Collection
- Finding Form: The Work of Richard Filipowski

Friends Online:

- Jeffrey Hoff
- Matthew Hockenberry

Figure 10. Screenshot showing the Billboard



Prayer and Bible Study

We have a half hour prayer followed by a one hour Bible study.This semester we are going to cover the books of Ephesians. We pray to the Lord to help us

W20-303 (move to W11-180 for Apr. 6 & 27 only)

5:30 PM - 7:00 PM

Figure 11. Screenshot showing the Billboard

4. Possible Applications

There are many possible groups that would use the placeMap project and benefit from its uses. The primary focus group for future use is universities, as the initial application was made for the Massachusetts Institute of Technology. Events on campus have multiple media on which they are conveyed, such as poster board, radio, websites, and emails. The placeMap application allows all events to be gathered together in one space, visually placing them at their respective locations. The project is set up to become an information exchange, not only of where and when events are happening, but of who is going and what they think about the event.

Aside from individual groups, major companies are highly interested in the project. Pepsi and Schlumberger are attracted to the idea that they could visually track vehicles, equipment, and people through one application, as well as use it as a medium for meeting scheduling. This interest shows the diversity of the possible applications this project can accommodate.

5. Conclusions

By looking at how a user would like to have a map tailored to his needs, placeMap contains a new type of map altogether. The user customizes the map by selecting friends to interact with, events to attend, and dates to view. In this way, the map wraps itself around the user, as opposed to the user conforming to the maps perspective.

Perspective skewing allows the user to feel more comfortable in the application environment, as it is more natural. Eliminating the typical top-down view of the world eases the user's transition to the map by making them feel as if they are really looking *out* onto the world as opposed to *down* on it. Changing the color or showing a "buddy" peeking over a location marker allows the user to passively receive information. Random images from web searches assist in removing monotony from the program. All of these minor details enhance the user experience by customizing their session to their needs. The most exciting conclusion is that placeMap shows the ability for user-based mapping to work and be accessible through many types of systems. The many possible applications realized and suggested attest to this conclusion.

6. Future Work

The placeMap project is now a good platform for user research on human-place interaction, human perception of place, and how place affects human decision making. Possible future work should include human studies in these areas. Also, broadening the aspect view of the project would probably prove to be beneficial to the study. Currently, the project is user-based, but what would happen if it became group-based? Integrating GPS would enable real-time tracking of objects through the world, however because of current licensing restrictions, Yahoo! does not allow this.

Natural language processing is an extremely useful tool to aid in search of information. Keyword search, a very commonly used technique, fails when the user expresses an idea without using the exact words as the author. A component is currently being developed to process the user's query in a way that aids their search. In the example search "food at the Media Lab," the user is obviously looking for "food" at a certain location, "Media Lab." By using the comprehensive possibilities that natural language processing possesses, placeMap aims to enhance the ability to search through events.

7. Acknowledgements

I would like to thank several people for their help with this project. First, Shawn Sullivan connected me with the Context Aware Computing group at MIT's Media Lab, where I did my project. Secondly, Matthew Hockenberry was a great friend and mentor throughout the entire project. I am truly privileged to be able to work with him on the placeMap project. Lastly, Professor Ted Selker trusted me enough to give me responsibilities over such big project. For that, his guidance, and all the opportunities with which he has provided me, I thank him.