AWEsome Wall
Team Zamboni Drive-by
Patrick Day  Paul Fisher  Monica Hirst
Brian Murphy
The MITRE Corporation
Marc Cannava, Doug Phair
Faculty Coach
J. Scott Hawker

Project Overview

The MITRE identification and point-of-presence system (renamed AWEsome wall) is a product that identifies nearby employees and displays personalized information in the form of a dashboard. The dashboard allows the user to customize their experience by selecting from a list of gadgets to use. Gadgets range from the personal, like digital photo frames, to the productive, like to-do lists. The system is intended to be implemented in temporary offices, known as hotel offices, with the goal of making these generic offices feel homey and comfortable to work in.

The gadgets are implemented using OpenSocial, a web API based on the Google Gadget framework. This system supports the use of pre-existing MITRE gadgets as well as specialized AWEsome wall gadgets. The gadgets aim to be “ambient” and allow the employee to quickly gather the information they need with little to no interaction with the wall.

The system will include several hardware elements. Hotel office users are currently identified by a smartcard reader in the room. Also, an office wall will be the host of an interactive whiteboard which will be displayed via projector.

Basic Requirements

1. User Identification
   a. Description and policy
      i. Upon entering an office, the user of the hotel office will be identified. The user will place their identification badge in a smartcard reader.
   b. Response sequences
      i. User enters office
         1. User places identification badge in smartcard reader
         2. User is identified by system and is logged in, launching AWEsome wall and picture frame
   c. Functional requirements
      i. AWEsome wall software and hardware must be present in office and running
      ii. User is a registered MITRE employee
2. Custom AWEsome Wall Launch
   a. Description and policy
      i. Upon being identified, the user’s personal dashboard will be launched on the screen in the room
   b. Response sequences
      i. AWEsome Wall is notified of identification of user in office
         1. System launches browser to wall portal page
         2. Portal SSO (Single sign on) authenticates the user
         3. Wall is launched with pre-saved gadgets by the user
   c. Functional requirements
      i. AWEsome wall software and hardware must be present in office and running
      ii. User is a registered MITRE employee
      iii. AWEsome Wall must have access to the internal MITRE network

3. Log user out from dashboard
   a. Description and policy
      i. At the end of the work day, a user is logged out of their dashboard if their identification badge is no longer in the smartcard reader
   b. Response sequences
      i. User leaves the office for the day (identification badge removed from smartcard reader)
         1. Logout service runs at a TBD time at the end of the business day
         2. Dashboard logs out user and saves all information still open in the environment
   c. Functional requirements
      i. AWEsome wall software and hardware must be present in office and running
      ii. User is a registered MITRE employee
      iii. User must not be in office at time of logout service execution

4. Add gadget to AWEsome wall
   a. Description and policy
      i. A user can add a gadget from the library to their unique wall
   b. Response sequences
      i. User hits “Add Gadget”
         1. User is shown a library of gadgets to select from for their board
         2. User selects gadget and presses “Add to my wall”
         3. Gadget is added to user’s AWEsome wall
   c. Functional requirements
      i. AWEsome wall software and hardware must be present in office and running
      ii. User is a registered MITRE employee
      iii. User must not be in office at time of logout service execution

Constraints
Hardware Constraints:

- The sensor for identifying a user will be a smartcard reader, reading a user’s physical badge
- The AWEsome wall will be projected onto a whiteboard interactive display, using a Mimio electronic whiteboard suite
- The picture frame to display user’s photos will be a Kodak picture frame

Security:

- The system must accommodate an active directory structure such as LDAP for identifying users

Other than these minimal constraints, the team was encouraged to tackle the project problem with “an open mind”.

Development Process

For our development process, we created our own process. It was influenced by the Open Unified Process (OpenUP). It applies iterative and incremental approaches within a structured lifecycle. We weren’t working with a full work week (i.e. working 40 hours a week on the project), so we met twice a week on Tuesdays and Thursdays. We supplemented the breaks with increased online communication and smaller scoped increments.

This process was approved by the sponsor, as they enjoyed the two week iteration timeframe. It allowed for rapid feedback on the product and prevented the project going down the wrong path. This process was not mandated by the sponsor.

Our process had three different layers: task blocks, an iteration lifecycle and the overall project lifecycle. Each task block included personal or paired efforts and lasted for a period of 2-3 days. Iterations lasted 2 weeks and included a number of tasks throughout the 6 task blocks. At the beginning of each iteration the general tasks are planned. Meetings took place at the beginning of
each iteration for planning, and continued through the iteration. The project lifecycle lasted a total of 20 weeks and was split into 4 iteration phases, Inception, Discovery, Construction, and Transition.

**Task Blocks**
- **Task Block 1** - Sunday & Monday
- **Task Block 2** - Tuesday & Wednesday
- **Task Block 3** - Thursday, Friday, & Saturday

**Inception phase**

As the first of the four phases in the project lifecycle, Inception is about understanding the project scope and objectives.

**Main objectives**
- Initiate Project
- Plan and Manage Iterations
- Identify Requirements
- Agree on Technical Approach

**Completion goal**
- 2 Iterations
  - January 9th, 2011 (accurately achieved)

**Discovery Phase**

This is the second of four phases in the project lifecycle, when technologies are explored and initial designs are addressed.

**Main objectives**
- Refine Project Schedule
- Explore Technologies
- Design Architecture
- Refine Documentation
- Develop Testing Plan
- Deliver Interim Presentation

**Completion goal**
- 3 Iterations
  - February 18th, 2011 (actual completion: February 25th, to allow additional work over spring break)
**Construction Phase**

As the third of the four phases in the project lifecycle, Construction focuses on refining design, implementation, and testing of functions to develop a complete system.

**Main objectives**

- Refine Project Schedule
- Refine Documentation
- Subsystem Implementation & Refinement
- Testing Solutions

**Completion goal**

- 3 Iterations
- April 17th, 2011

**Transition Phase**

Fourth and final phase in the project lifecycle, Transition focuses on deploying software to users and ensuring that their expectations about the software were met.

**Main objectives**

- Refine Project Schedule
- Sponsor Delivery
- Return Project Resources
- Deliver SE Department Documentation
- Final Presentation

**Completion goal**

- 2 Iterations
- May 13th, 2011

**Project Schedule: Planned and Actual**

**Planned Schedule**

As mentioned above discussing the Open Unified Process, we broke down our schedule into four blocks. The first two phases (Inception and Discovery) consumed the first ten weeks of the project. The final ten weeks were consumed by the construction and transition phases, more heavily so on the construction phase. These four phases are our major milestones. The two week iterations in these phases were also minor milestones.
Two gadgets were designated to be developed each increment, with two team members dedicated to each gadget. This included initial design and wireframe phases, with an expected prototype at the end of that two week cycle. The planned schedule in its entirety is below:

<table>
<thead>
<tr>
<th>Date</th>
<th>Item Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/6/2011</td>
<td>Project Plan Delivery</td>
</tr>
<tr>
<td>1/8/2011</td>
<td>SRS Delivery</td>
</tr>
<tr>
<td>1/9/2011</td>
<td>Inception Phase Complete</td>
</tr>
<tr>
<td>1/25/2011</td>
<td>Customer Visit</td>
</tr>
<tr>
<td>1/26/2011</td>
<td>In Person Requirements Walkthrough</td>
</tr>
<tr>
<td>2/5/2011</td>
<td>Architecture Document Delivery</td>
</tr>
<tr>
<td>2/18/2011</td>
<td>Test Plan Due</td>
</tr>
<tr>
<td>3/8/2011</td>
<td>Customer Visit</td>
</tr>
<tr>
<td>3/18/2011</td>
<td>Awards and Achievements Gadget Due</td>
</tr>
<tr>
<td></td>
<td>While You Were Out Gadget Due</td>
</tr>
<tr>
<td>3/31/2011</td>
<td>Customer Demo</td>
</tr>
<tr>
<td>4/1/2011</td>
<td>OSEC Modifications Due</td>
</tr>
<tr>
<td>4/15/2011</td>
<td>Weather Gadget Due</td>
</tr>
<tr>
<td>4/15/2011</td>
<td>Whiteboard Gadget Due</td>
</tr>
<tr>
<td></td>
<td>Photo Frame Development Due</td>
</tr>
<tr>
<td></td>
<td>Calendar Gadget due</td>
</tr>
<tr>
<td>4/20/2011</td>
<td>Project Poster Due</td>
</tr>
<tr>
<td>4/29/2011</td>
<td>Traffic Gadget Due</td>
</tr>
<tr>
<td></td>
<td>Virtual Window Gadget Due</td>
</tr>
<tr>
<td></td>
<td>Construction Phase Complete</td>
</tr>
<tr>
<td></td>
<td>Solution Delivered</td>
</tr>
<tr>
<td></td>
<td>Project Poster Day</td>
</tr>
<tr>
<td>4/30/2011</td>
<td>Final Presentation Draft</td>
</tr>
<tr>
<td>5/5/2011</td>
<td>Deliver Final Presentation</td>
</tr>
<tr>
<td>5/13/2011</td>
<td>Transition Phase Complete</td>
</tr>
<tr>
<td>5/14/2011</td>
<td>Technical Report Due</td>
</tr>
</tbody>
</table>

**Actual Schedule**

There was some slippage in our schedule. This was attributed to requirements changes early in the spring quarter. The customer wanted the container development to take over as compared to delivering individual gadgets. Without a stable and unique container, we were told the project would not succeed inside MITRE. We were able to negotiate with our customer and drop the calendar gadget as a deliverable. MITRE stressed that they have people inside who have built gadgets on this platform and can deliver these gadgets later. The main thing was delivering the “whole package”.

Because of these changes, our Awards and Achievements and While You Were Out gadgets required a schedule delay. We had begun developing these gadgets prior to the requirements.
change, and a redesign was necessary to meet the new needs of the container. The gadgets that followed stuck very close to their delivery schedule.

We benefitted from MITRE resources as the project progressed. This helped us stay on schedule as well as ahead of schedule on some items. Since MITRE’s OpenSocial Container was fairly new to the open source community, there was little documentation when we ran into errors. MITRE’s internal development team was very helpful and speedy in helping us when we ran into trouble. Notably, Jesse Ciancetta and Matt Franklin were vital to the container’s success. One item to note was that without the personal connection with both Jesse and Matt, we would have never been able to deliver the product we did. The documentation in the OpenSocial community was insufficient for our needs. Also the help of John Ursino in poster creation gave us some extra time to focus on development.

System Design
Our design was driven heavily on the requirement that the user must have the same experience no matter where they are connecting from. Obviously, a web based model was optimal. There are major components to our architecture:

OpenSocial
We chose the OpenSocial design as it was pre-defined by the OpenSocial framework. All components (gadget server, Apache Shindig, OSEC), were hosted on our project server. We used Shindig to host the OpenSocial Enterprise Container, as well as hosting our gadget configurations. When adding a gadget to the container, a valid URL pointing to the gadget specifications are required.

We considered alternatives for gadget hosting, but in the end found this structure was vital for ensuring successful integration with MITRE’s infrastructure. They already had an OpenSocial environment deployed and running gadgets. We considered alternatives for OSEC, such as Partuza, a gadget rendering container in PHP, however MITRE’s written OSEC (released to the open source community), provided additional security features such as Single-Sign-On (SSO), as well as greater stability.

AWEsome Wall Client
The AWEsome wall client represents the physical machine in the hotel office. This is the host for a variety of services and clients.

First and foremost, the OpenSocial client lives on this client. This is what the user sees and interacts with on the wall. This client was placed on the physical machine as compared to a standalone display, since we have the benefit of domain authentication on the machine, which allows us to utilize Single-Sign-On (SSO). This was the most obvious solution, since the OpenSocial client lives within a browser. We considered having the client on a standalone machine in the room, however authentication became a concern. Also, the client is then protected when a user locks their computer.

The picture loading service also lives on this machine. This was due to the fact that the digital picture frame connects to a folder of a machine on its wireless network. This allows the frame to be tied to the physical machine in the room, regardless of who is logged in. When a user logs in, we can populate a static folder with that user’s photos, and have them appear on the frame. We
considered a variety of solutions for delivering photos to the user’s desktop; however in the end this seemed to be the most practical solution. Kodak offers a service to upload photos to the frame via e-mail, however in our case this solution would be too slow, as it took about five minutes for a single image to be delivered to the frame. In trying to deliver a user’s entire photo album this would be impractical.

Finally, the Mimio capture and interaction services exist on this client. Since this drives the interaction with the OpenSocial client, these two components must exist on the same machine. We considered alternatives to Mimio, such as SMART interactive touch systems, and touch LCD systems. The benefit of Mimio is its portability. The Mimio sensor can be mounted to a whiteboard and calibrated within minutes. This could allow customization by the user as to where the sensor is placed.

**Kodak Picture Frame**

The physical picture frame does not have many constraints. We need it to map to a folder on a computer, which must be on the same network the frame is connected to. This requires the physical frame and the AWESome wall client to be on the same network. We considered a few other frames; however this hardware was supplied by the client and provided all the functionality we needed.

**Identification Sensor**

The identification sensor is able to identify a user based on their badge. This sensor is not required to be connected to a computer, so it can be broken into its own category. The sensor client is the physical sensor device. This is where the user taps their badge to be identified. That code is then sent via TCP to the sensor identification server. The server then determines who has tapped their badge by converting the badge ID to an associated person.

This component was not completely integrated into the project. MITRE is working to grant domain authentication upon inserting their badge into a card reader. Our expected deliverable is a service that identifies a person based on the badge ID that was swiped on the reader. This was done over a TCP connection. We encountered hardware failure within the last few weeks in the project. However, we were able to deliver the badge identification service and PIN authorization process. Without functioning hardware we could not successfully demo the login process at the end of the project. With functional hardware our implementation will meet MITRE’s needs.

We considered a variety of identification solutions in the discovery phase of the project. Items included RFID, RuBEE, and facial recognition. However, as this component of the project became overshadowed by the AWESome wall and photo frame, we did not have time to explore all of these identification solutions, as they would consume the bulk of our development time.

**Process and Product Metrics**

For our process metrics, we were able to observe many statistics provided to us by Trac, our time tracking tool, as well as Handshake, the collaborative environment with our sponsor. In Trac, we monitored the following metrics:

- Tasks per increment (at time of report delivery)
  - Inception:
    - 53 total hours, 56 estimated (95% accuracy)
20 work tickets

- Discovery:
  - 96 total hours, 119 estimated (81% accuracy)
  - 46 work tickets

- Construction:
  - 201.3 total hours, 315.4 estimated (63.4% accuracy)
  - 28 work tickets

- Transition:
  - 40 total hours, 48 estimated (83% accuracy)
  - 10 work tickets

These results were able to help us identify how high our level of effort was on the project. Being able to monitor the hours worked per iteration allowed us to see how hard we were working on those goals.

Additionally, we were able to monitor our handshake metrics:

- 33 files uploaded, with 17 comments on these files (around 1.65 files uploaded per week)
- 45 discussion topics, with 158 replies in these topics (10.15 discussion posts per week)

These Handshake metrics helped us see how often we were interacting with our sponsor outside of our team meetings. At one point our discussion metrics declined a little, hinting we should be interacting with our sponsor more. Upon applying this observation, we saw our requirements gathering complete faster than we would have expected without the additional interaction between meetings.

For our product metrics, we initially targeted many metrics around our sensor identification time and success rate. However, as the project progressed, this metric was not as important, as the sponsor approved identification via Single-Sign-On (SSO), which does not require the sensor. However, we did target the following metrics:

- Hotel office load time
- Digital picture frame load time upon login

**Product State at Time of Delivery**

Looking back upon what was planned initially during our discovery and inception phases, our delivered product mutated a bit during our construction. Our sponsor heavily urged that they did not want the AWEsome wall to resemble a portal, similar to iGoogle. Since MITRE already has a similar portal structure established internally, this wall had to have a unique look to it. Early in our second quarter of work, we shifted our schedule from primarily gadget development towards modifying the gadget container to have a custom, ambient feel with the hotel office.

Unplanned features that were added were focused around the gadget container. Building the layout of the container to not be a portal gave us some additional requirements. Also, having multiple views for each gadget was unforeseen. For example, our weather gadget took on an “ambient” view when in a small mode, and then took on a more detailed view when full screened. The biggest part of the gadget development was when specific information is delivered to the user, and what view that should fall into.
Planned features that are missing include the calendar gadget and domain authentication with the badge reader. As mentioned above, we restructured our schedule to accommodate gadget container modifications. The badge reader change was due to exploration results within MITRE, determining this would be out of our reach in the time we had for this project. Our sponsor approved the schedule change, and was ok with removing gadgets from the schedule.

**Project Reflection**

Overall, we see this project as a success. We delivered a functional hotel environment, where users can create a custom office feel that will travel with them when they change office locations daily.

We feel that we could have been more proactive in our requirements gathering phase. When the project began, it was very open-ended. A major piece of it was brainstorm and research based. This delayed our development until spring quarter. If we were able to begin development earlier, we could have had more time for additional gadget development.

In terms of process, we feel the modified version of the Open Unified Process met our needs well. It gave us the flexibility Scrum could not, since Scrum would have required daily standup meetings. As college students, this seemed like an unattainable requirement. Also, task monitoring in Trac allowed us to see what we were required to deliver for that iteration, which was very helpful.

Our sponsor interaction was vital to moving this project to a final product. As mentioned earlier, there were no major requirements set when the project began. Narrowing down the sponsor’s ideas into a final, unified idea took some time. With only having one hour of face-to-face time a week, Handshake was vital to progressing the project ideas into a product.

Although the delivered product was not what was initially described in the project proposal by the customer, we feel we successfully elicited new requirements, adapted to requirements change, and delivered a working product the customer wanted.

**References**

**Contact Information:**

**Client**
Doug Phair: dphair@mitre.org
Marc Cannava: mcannava@mitre.org

**Team**
Patrick Day: daypatrickb@gmail.com
Paul Fisher: paul@tdi.us
Monica Hirst: dragongirl777@gmail.com
Brian Murphy: brianmurphy425@gmail.com

**Faculty Coach C**
J. Scott Hawker: jshvse@rit.edu
Related Project Items

1. AWE: Agile Workspace Environment
2. The MITRE Corporation: [http://mitre.org](http://mitre.org)
5. Handshake: [http://handshake.mitre.org](http://handshake.mitre.org)
9. iGoogle: [http://google.com/ig](http://google.com/ig)
10. Kodak PULSE Photo Frames: [https://www.kodakpulse.com/](https://www.kodakpulse.com/)