Demonstrating Evacuation Algorithms with Mobile Devices using an e-Scavenger Hunt Game

Jesse Szwedko, Callen Shaw, Alexander G. Connor
Alexandros Labrinidis, Panos K. Chrysanthis
Advanced Data Management Technology Laboratory
Department of Computer Science, University of Pittsburgh, Pittsburgh, PA 15260, USA
jjs86@pitt.edu, cts8@pitt.edu, agc7@cs.pitt.edu, labrinid@cs.pitt.edu, panos@cs.pitt.edu

ABSTRACT

Casualties in emergency situations are often caused by panic and in cases where building evacuation is required, they are often caused by a disorganized evacuation. This has motivated us to design a two-layer indoor evacuation system that takes advantage of two technologies all people carry on them, namely, cellular phones with cameras and RFID cards. The proposed system integrates QR Code and RFID-based positioning with a routing system with mounted terminals and displays for guiding people with RFID tags out of a building. People with mobile devices with cameras use an application that resolves QR codes into web addresses that point to dynamically generated evacuation instructions. As a proof-of-concept, we have implemented this system with commercially available tools and components as an e-scaveger hunt game which uses our novel evacuation (routing) algorithm to guide players around a building visiting different locations in a load balancing manner. In this demo, we are planning to deploy this e-scavenger game and the participants would be able to follow the progress of the game (evacuation) through a system monitor dashboard.

1. INTRODUCTION

Building evacuations are a common occurrence, so a system which guides users to safe exits could be of great assistence. Currently, almost all buildings have lighted exit signs and evacuation maps at specific locations (at intersections or near doors), but these are unable to provide information such as whether the exit is inaccessible or overcrowded which can lead to disorganized and potentially dangerous evacuations. A better idea might be to somehow direct the evacuees through a location-aware system. Ideally, this system would make use of GPS to help users to identify their location and how to proceed to nearby exits. However, an indoor navigation system cannot currently rely on GPS information because the information is often incomplete or inaccessible in indoor scenarios as well as being traditionally only two dimensional which is of little use in multi-floor buildings.

Instead, we have designed a system which dynamically guides users to safe exits in a load balanced manner, without requiring specialized monitoring equipment. The system was designed to take advantage of two technologies almost all people carry on them - cellular phones and RFID cards. Many companies already require electronic ID cards, and these would be used in conjunction with display screens showing an evacuation guide. If this system failed for any reason (due to damage or a lost internet connection), a second, more robust system coexists to facilitate building escape. QR (Quick Response) Barcodes, already used to form a rich cyber-physical connection to objects, places, or products (a number of which uses are identified in [7]), would be affixed at various locations in the building. A user can then use already existing QR Code software for their phone to receive evacuation directions via their mobile web browser.

Our system uses its two subsystems to reliably and safely direct people to the least crowded and safest exit based on previous queries (from interactions with the RFID and QR frontends) to the system, and has the ability to remove exits that have become unsafe. The users can access the RFID system quickly, by holding their RFID card close to a reader. This sends a request to the backend database system, and their personalized instructions appear on a display screen. This system's functionality can be hindered in the event of power failure to the readers or a loss of connection to the backend. For our system to work in a disaster scenario, QR Codes can be scanned by existing software to load a unique webpage, which is encoded into the QR Code. This will load a webpage on any internet-enabled cell phone showing them the way to exit. It should be noted that while the QR Code system can serve as a back up for the RFID system, it operates concurrently with RFID Code systems and people can use them interchangably since both systems utilize the same backend server. Since our RFID and QR Code systems utilize a shared server, the load balancing evacuation (routing) algorithm is able to intelligently reduce bottlenecks for all users. By utilizing both frontend systems our design operates in a variety of conditions without performance failure, and in an evacuation scenario the lessened crowding afforded by load balancing will ensure users can exit quickly and safely.

Both QR Codes and RFID tags have been used in other location-based services to provide individual users personalized directions ([2], [6]), but our system is unique in how it coordinates both services to provide dynamic location awareness to multiple users.

As a proof-of-concept, we designed an electronic scavenger hunt which demonstrates the principles of our proposed evacuation system, as described above. It was deployed during our departmental open house (Computer Science Day 2009) to guide guests around the building to visit various vendor booths. The booths acted as our exits, and the load balancing algorithm was employed to ensure guests were distributed evenly across all booths. Some booths were given both an RFID reader and a QR Code, and others were equipped with QR Codes only. Feedback from system usage was utilized predict which areas of the building were heavily crowded, and users received their scavenger hunt assignments with respect to this crowding.

Contributions

- A novel load balancing algorithm for traffic flow management.
- A way to seemlessly integrate pre-existing and relatively commonplace technologies (RFID and QR codes) in a way to efficiently direct traffic flow of large number of people to desired locations.
- A scavenger hunt game that effictively demonstrates the system's capacity to act as an evacuation catalyst.

To demonstrate the effacacy of our system during the conference, we will mimic the scavenger hunt application which was successfully deployed for Computer Science Day 2009. The application will route users from location to location according to the load balancing algorithm described below in Section 3 to demonstrate both RFID and QR frontends for the system.

2. SYSTEM ARCHITECTURE

The scanvenger hunt application's architecture is structured as three interconnected subsystems – two frontend and one backend – which make up the overall Scavenger Hunt. This is illustrated in Figure 1.

RFID Subsystem The first frontend subsystem, called the RFID system is comprised of two tiers. The hardware (which is considered as the first tier) is comprised of RFID readers that are strategically placed at specific locations that the administrator wishes the users to visit as well as the RFID cards which are distributed to each user. When the user visits a RFID reader they swipe their RFID card over it and an event is triggered in the java application, being the second tier of which the user interacts with, running on a display terminal connected to the reader is triggered. The display terminal then displays whether or not the user went to the correct location (in the case of the first visit this check is skipped) as well as their next destination. The user's next destination is chosen by the algorithm running on the backend subsystem described in Section 3.

QR Code Subystem Similarly the second frontend subsystem, referred to as the QR Code system, is also comprised of two tiers. The bottom tier of the QR Code system consists of a collection of print outs containing unique QR Codes, that can be read by many smart phones out-of-the-box, that are placed at the locations the users are desired to visit. Encoded in the QR code is a web address, common to all of the locations, which links to the scavenger hunt website, along with a unique GET variable which is linked to each location uniquely in the database. The URL, with the GET variable, is meant to be kept short enough such that the user can type into their internet capable phone if it does not carry the required software to process the the QR Code.

The second tier of the QR subsystem is a simple mobile-friendly web frontend accessed through the mobile phone. When a user either scans a QR code with their phone or enters the associated URL into a browser for the first time the user is authenticated using simple scheme developed both to use no personally identifiable information as well as be easy to use on mobile devices. This scheme involves the user first entering the unique number which is printed on their RFID card followed by choosing the name on the card from a list of names.

When the user subsequently accesses a URL encoded in one of the QR codes they are taken to the encoded URL that displays information relevant to the location that they are currently at. In the case of the scavenger hunt application this website displays one of three things:

1. That the user has visited the correct location along with a question relevant to the location. The displayed question is to ensure that the user does not game the system by running from location to location rapidly. When they answer this question (either correctly or incorrectly) they are awarded some number of points.
2. That the user has visited the incorrect location.
3. Statistics relevant to the logged in user such as the history of locations visited or how far they have progressed in the scavenger hunt.

After either of these scenarios, the user is directed to a new location by the load balancing algorithm.

Backend Database System Both the RFID system and QR Code system are backed by the same backend subsystem which consists of a MySQL database sytem that is accessed directly by the RFID and QR Code subsystems. This database stores the locations as an edge-weighted graph for use with the load balancing algorithm. The algorithm is written as a stored procedue that is called by the two frontend subsytems described above. This common infrastructure allows the users to switch between the two fontend systems with ease, but the independent nature of the frontend systems ensures that, should either the QR Code or RFID system go down, the users can continue using the other.

3. EVACUATION ALGORITHM

Description

Existing approaches to the evacuation problem use a network flow model of the building to be evacuated [1, 5, 4]. There are several metrics in the literature that such approaches optimize for, including maximum dynamic flow, quickest flow and earliest arrival. The first of these are classical network flow problems; earliest arrival is a metric that will minimize the spent in a building for any person.

Our approach is quite different in that it does not use network flow to model the passage of persons through a building, it uses a vertex-weighted graph. The data structure

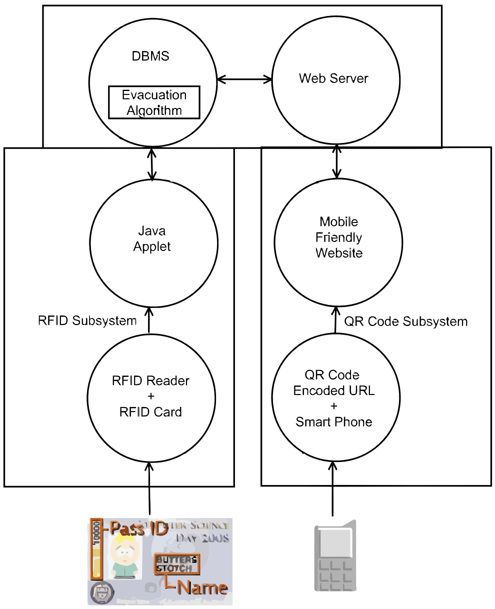

Figure 1: Representation of the system architecture. The two frontend systems are independent of one another, but backed by the same webdb system infrastructure.

is fairly simple; in our scavenger hunt demonstration, each booth was represented by a vertex. The edge costs between a pair vertices reflected the time required to travel from the start vertex to the end vertex. The weight at each vertex is then a measure of how many people have been instructed to go to that vertex.

The overall cost of traveling from one booth to another was determined by summing the cost of the shortest path from the start to end vertex and a function of the weight of the end vertex. With this distance function, both the cost of walking to a booth and waiting to use the terminal at that booth were considered for the user.

Routing in this model was done by a nearest neighbor query; the query picked the vertex such that the cost of traveling to the vertex plus the cost of visiting that vertex was minimized. The cost of traveling to a vertex was computed as the shortest path distance to the vertex and the cost of visiting was a function of how many people have already been sent to the vertex. In this way our approach finds routes that reduce the time a person spends in the network and prevents bottlenecks by directing people away from over-crowded vertices.

The evacuation algorithm takes as input the vertex that the user is currently at. It then explores the network starting from that vertex using Djikstra's single-source shortest path algorithm [3]. Once this is done, the start vertex and the vertices that the user has already visited earlier in the scavenger hunt or evacuation are removed from the candidate list. The current weight of each vertex is then added to its shortest path cost, and the minimum of these is returned. The vertex weight for the returned nearest neighbor is then increased, reflecting the increase in traffic directed to it.

In the evacuation scenario, there will need to be two types of vertices; exit vertices and intermediary vertices. Intermediary vertices are used in computing routes through the building but will never be returned by the evacuation algorithm as they are not valid end destinations. Exit vertices are the only kind returned by the algorithm, and their vertex weights are maintained by the system so as to reflect how crowded they are. In the scavenger hunt game, all vertices were like exit vertices in that they were found by the nearest neighbor search and their vertex weight was maintained by the system.

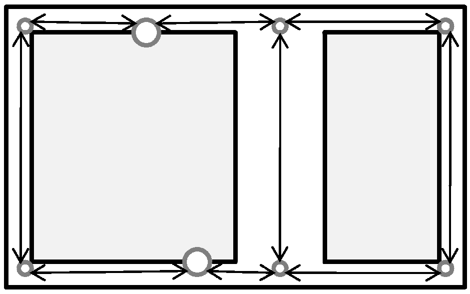

Figure 2: A building map and its evacuation graph. The large circles indicate exits (stairwells) and the small circles indicate intermediary points; there is an RFID terminal and QR code at each point.

The evacuation algorithm, when called from an intermediary vertex, will return the user's nearest exit and the next vertex on the shortest path to that exit. A terminal at that vertex will display this information and also give an indication of how urgently the user should proceed. This urgency information could easily be gathered from an outside source, such as a sensor network or monitoring system. When the algorithm is called for the first time, the user is assigned to the returned exit and that exit's vertex weight is increased. Subsequent calls from intermediary vertices will return a different exit only if the assigned exit is inaccessible; these calls will also compute the shortest path dynamically, so if a hallway becomes impassible after the initial query a user will not be sent on a path through that hallway.

Evaluation

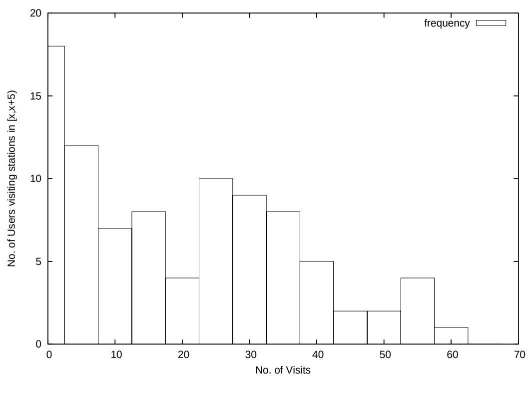

Figure 3: Frequency distribution of visits per user for the scavenger hunt. Our scavenger hunt catered to a variety of participation levels.

We had 92 users total and 32 booths. 19 of these booths had an RFID reader and QR code; all booths had a QR code. Of the 92 users only 24 had QR-enabled devices, and all users had an RFID card. In the evacuation scenario, all users would have both an RFID card – perhaps in the form of an identification badge – and possibly QR-enabled device. Also in this scenario there would RFID and QR code terminals throughout the building.

During the CS Day 2009 deployment we recorded 2,523 visits during the scavenger hunt game. To measure user participation, we took a frequency distribution of visits per user in Figure 3. As the number of visits increases, the number of users making visits in that range decreases. There were two outliers: One user made 182 visits and another made 367. We feel that this broad range of participation is because our users had a range of interest in the game. Some were just curious about it and others were committed to making as many visits as possible.

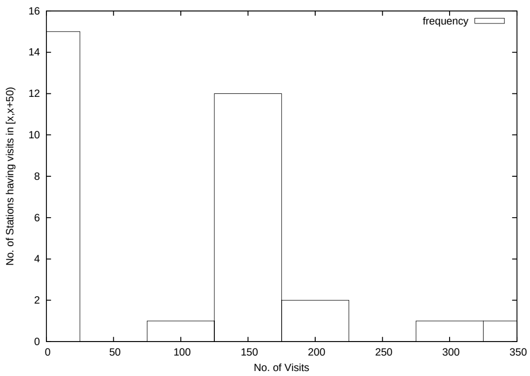

Figure 4: Frequency distribution of visits per booth for the scavenger hunt. Although the distribution is bimodal, the booths that received more than a couple visits received about the same number of visits.

Our approach worked remarkably well in terms of balancing load during our demonstration. One sees in Figure 4 that the distribution is bimodal; many booths received few visits – we suppose that this is because they did not have RFID readers – and 12 of the 19 that did have RFID readers saw between 150 and 200 visits. There were two booths that saw more than 300 visits; we believe that the users mentioned above who made more than 150 visits went back and forth visiting these two booths.

4. DEMONSTRATION SETTINGS

Equipment

For the conference demo we will set up a playable scavenger hunt game, replicating the system which was deployed at CS Day 2009. To participate, a user will receive an RFID card with a printed number and avatar on it. We will bring about 100 of these cards for attendees. The system can be interacted with by using only the RFID frontend, the QR web frontend, or a combination of the two.

For demonstration of the RFID frontend, we will bring three or four laptop computers which have USB RFID tag readers attached to them. Users who scan their cards at the correct RFID scanner will be directed (through on-screen instructions) to their next destination. We will also bring an additional laptop to provide connection to the internet as well as running a system monitor dashboard.

Participants with internet-enabled cell phones can access the web frontend as well. The equipment needed for this is simply a number of printed QR Codes to be placed throughout the conference. We will bring two phones with QR Code software installed as well, to demonstrate the system for users who cannot or do not wish to use their own phone.

Our two frontends will both access the internet to communicate with our backend database system. The RFID scanners are connected to machines which will access an available internet connection. Since the QR Code frontend is designed for mobile browsers, the web server will communicate with the database as users interact with it. If an internet connection is unavailable, we will attempt to set up an ad-hoc network with a local database system that will allow for the use of the RFID subsystem as well as the QR Code system, but only with WiFi enabled phones since the webserver won't be accessible from the outside (the on-hand cell phones for demonstration will fall in this category).

Demo plan

We envision that all participants will bring a poster to display, and that these will be arranged throughout multiple corridors. We will place printed QR Codes in close proximity to the posters, and use these easily recognizable landmarks to guide participants. The system will serve as an analogy for the evacuation scenario by directing users through these corridors, as if in an emergency.

Prior to the demonstration, our database will be initialized with a graph of the QR Code and RFID locations. This graph is used by the load balancing algorithm to ensure that assignments are distributed evenly among these locations. After an assigned location is visited, the user's next assignment is generated taking system load into account. Our cyber-physical system will distribute users evenly to prevent problematic crowding and prove the effecacy of our design.

We will have one laptop set up which will display a dashboard that shows the real-time changes within the system, load at the different nodes, and combined metrics. This information will demonstrate the load balancing algorithm's ability to route users effectively.

A user will be required to visit a certain number of stations to "'complete"' their hunt, but the system will continue to generate assignments indefinitely so the it can be used as long as desired.

5. REFERENCES

[1] L. G. Chalmet, R. L. Francis, and P. B. Saunders. Network models for building evacuation. *Fire Technology*, 18(1):90–113, 2 1982.

[2] Y.-J. Chang, S.-K. Tsai, Y.-S. Chang, and T.-Y. Wang. A novel wayfinding system based on geo-coded qr codes for individuals with cognitive impairments. In *Assets '07: Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, pages 231–232, New York, NY, USA, 2007. ACM.

[3] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[4] B. Hoppe and E. Tardos. Polynomial time algorithms for some evacuation problems. In *SODA '94*, pages

433–441, Philadelphia, PA, USA, 1994. Society for Industrial and Applied Mathematics.

[5] Q. Lu, Y. Huang, and S. Shekhar. Evacuation planning: A capacity constrained routing approach. *Intelligence and Security Informatics*, page 959, 2003.

[6] F. Lyardet, D. W. Szeto, and E. Aitenbichler. Context-aware indoor navigation. pages 290–307, 2008.

[7] E. Rukzio, A. Schmidt, and H. Hussmann. Physical posters as gateways to context-aware services for mobile devices. pages 10–19, Dec. 2004.