Demonstrating an evacuation algorithm with mobile devices using an e-scavenger hunt game.

CONFERENCE PAPER · JANUARY 2009

DOI: 10.1145/1594139.1594154 · Source: DBLP

5 AUTHORS, INCLUDING:


Alexandros Labrinidis
University of Pittsburgh


Panos K. Chrysanthis
University of Pittsburgh

Demonstrating An Evacuation Algorithm with Mobile Devices using an e-Scavenger Hunt Game (Demo Paper)

Jesse Szwedko, Callen Shaw, Alexander G. Connor
Alexandros Labrinidis, Panos K. Chrysanthis
Advanced Data Management Technologies 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-scavenger hunt game which uses SCAVY, 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.

Categories and Subject Descriptors

H.2.4 [**Systems**]: Query processing; H.2.2 [**Physical Design**]: Access methods

General Terms

Algorithms, Design, Management

Keywords

RFID, QR-Code, evacuation, vertex weight, data management, mobile computing, nearest neighbor, navigation

1. INTRODUCTION

During a building evacuation, great care must be taken to remove as many people from the building as safely as possible. 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 real-time 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 a global positioning system (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: Internet-capable mobile phones, which are fairly popular, and radio-frequency identification (RFID) cards, which are very common in corporate environments and are quickly becoming ubiquitous in the mainstream. RFID cards and readers are used in conjunction with display screens showing an evacuation guide. QR-Code (Quick Response Barcode) signs and a mobile-friendly web application accessed via QR-Codes shows mobile phone users the evacuation instructions. QR-Codes are already used to form a rich cyber-physical connection to objects, places, or products (a number of such uses are identified in [8]).

The core of the system is a database system that maintains information necessary for an evacuation and within which $SCAVY$, our proposed algorithm for guiding users out of a building in an emergency, is executing. At any given point, the system keeps track which exits and hallways are safe, removing any that have become unsafe; it also tracks which user is directed to which exit. In this way, at any point in an evacuation, users are given the best directions possible.

Two front-end subsystems support the two instruction dissemination approaches; specifically, the RFID-driven and the QR-Code-driven approach. Both RFID readers and displays as well as QR-Code signs are strategically placed at specific locations in the building such as at intersections and the middle of long corridors. Users within range of a RFID reader can access the RFID subsystem effortlessly and quickly. The reader sends a request for evacuation directions to the back-end database system based on the RFID position, and shows the directions on a display screen.

Users with Internet-capable mobile phones can access the QR-Code Subsystem by scanning a QR-Code with their phone and receive evacuation directions via a mobile web browser. QR-Codes encode web addresses (URLs) that are used to transmit an evacuation request. Proprietary QR-Code reading software is available for most smart phones, and an open source version is available that works on most phones that run Java ME [4].

The two front-end subsystems operate concurrently and can be used interchangeably since both utilize the same back-end database system. Besides flexibility in using the system, this makes the system more robust: The QR-Code Subsystem can serve as a fail-over for the RFID Subsystem in the case of building infrastructure failure and the RFID Subsystem can serve as a fail-over for the QR Subsystem should wireless connectitivity become unavailable. By utilizing both front-end systems our design operates in a variety of conditions without performance failure.

As a proof-of-concept, we designed an electronic scavenger hunt which demonstrates the principles of our proposed evacuation system, as described above. The algorithm and system used in the scavenger hunt is fundamentally the same as those proposed in the evacuation system with some subtle exceptions. Less powerful RFID readers were used during the game because people were expected to spend time near each RFID-enabled terminal as part of the scavenger hunt. The terminals gave written instructions for completing the hunt and also provided notices to each user, as did the web application.

The scavenger hunt system was deployed during our departmental open house – Computer Science Day 2009 – to guide guests around the building to visit various vendor booths. There were 92 participating guests (users) and 32 booths. The booths acted as our exits, and our load balancing algorithm was employed to ensure guests were distributed evenly across all booths (safe exists). All booths were equipped with QR-Codes and 19 of these booths were given also an RFID reader. Feedback from system usage was utilized to predict which areas of the building were heavily crowded, and users received their scavenger hunt assignments with respect to this crowding.

Both QR-Codes and RFID tags have been used in other location-based services to provide individual users personalized directions [2, 7], but our system is unique in how it coordinates both services to provide dynamic location awareness to multiple users.

Contributions

- A novel load balancing algorithm – called SCAVY – for traffic flow management.
- A way to seamlessly integrate preexisting 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 effectively demonstrates the system's capacity to act as an evacuation catalyst.

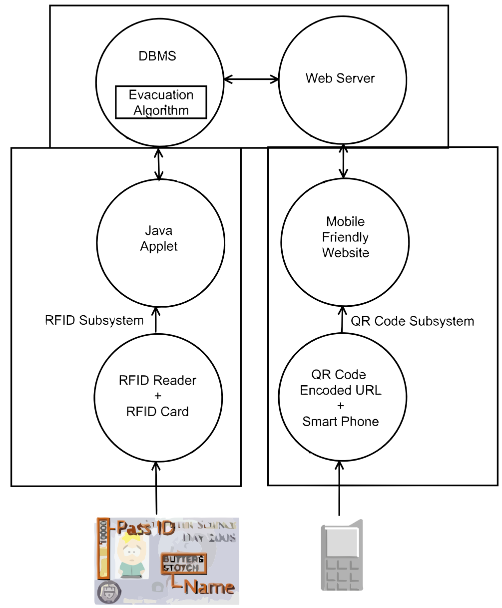

Figure 1: Representation of the system architecture. The two front-end systems are independent of one another, but backed by the same webdb system infrastructure.

To demonstrate the efficacy 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 SCAVY, the evacuation (routing) algorithm described in Section 3, to demonstrate both RFID and QR front-ends for the system. The details of this demonstration are described in Section 4.

2. SYSTEM ARCHITECTURE

The scavenger hunt application's architecture is structured as three interconnected subsystems – two front-end and one back-end – which make up the overall scavenger hunt. This is illustrated in Figure 1.

RFID Subsystem The first front-end subsystem, called the RFID Subsystem, has two tiers. The first tier, consisting of the hardware, is comprised of RFID readers connected to a network of stations. These stations, each connected to a display terminal, are strategically placed at various locations throughout the building. Also, this tier includes the RFID cards given to the users.

The second tier consists of a Java application running on each station. When a user visits a RFID reader he or she swipes his or her RFID card over it and an event is triggered in the second tier. The terminal connected to the station that drives the RFID reader 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 SCAVY, our evacuation (routing) algorithm, which runs on the back-end subsystem.

QR-Code Subsystem The second front-end subsystem, referred to as the QR-Code Subsystem, is also comprised of two tiers. The bottom tier of the QR Code Subsystem

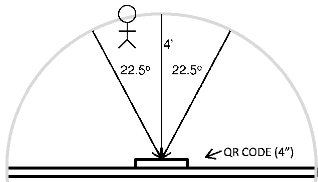

Figure 2: Users can access a QR-Code with their mobile phones while other evacuees have room to escape behind.

consists of a collection of signs containing unique QR-Codes, which can be read by many smart phones out-of-the-box, that are placed at the locations the users are desired to visit. We found that we could take a clear picture of a 4 inch wide QR-Code within a 4 foot radius up to 45 ° off of the perpendicular line-of-sight, although the current free software ("Barcodes" for iPhone) could resolve the address only within 22.5 ° (Figure 2).

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 HTTP GET variable which is uniquely linked to each location stored in the database. The URL, with the GET variable, is meant to be kept short enough such that the user can type into an Internet capable phone in the event that their phone does not carry the required software to process the the QR-Code.

The second tier of the QR-Code Subsystem is a simple mobile-friendly web front-end 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. Once authenticated, an HTTP cookie is placed on the mobile phone's web browser cache in order to avoid the to re-authenticate in the future.

Back-end Database System Both the RFID Subsystem and QR-Code Subsystem are connected to the same back-end subsystem which consists of a MySQL database that is accessed directly by the RFID and QR-Code subsystems. This database stores the locations as an edge-weighted vertex-weighted graph for use with the load balancing algorithm, SCAVY. SCAVY is implemented as a set of stored procedures that are called by the two front-end subsystems described above. This common infrastructure allows the users to switch between the two front-end systems with ease, but the independent nature of the front-end systems ensures that, should either the QR-Code or RFID Subsystem become disabled, the users can continue using the other.

In either case, when a user request is sent to the back-end one of three responses is given:

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 abuse 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.

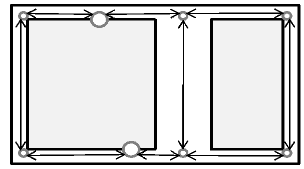

Figure 3: A building map and its evacuation graph. The large circles indicate exits (stairwells) and the small circles indicate intermediary points; there is an RFID terminal and QR code at each point.

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 the first two scenarios, the user is directed to a new location by the load balancing algorithm.

3. THE SCAVY ALGORITHM

Existing approaches to the evacuation problem use a network flow model of the building to be evacuated [1, 6, 5]. There are several metrics in the literature that such approaches optimize, including Maximum Dynamic Flow, Quickest Flow and Earliest Arrival. The first two 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 is fairly simple; in our scavenger hunt demonstration, each booth was represented by a vertex. The edge costs between a pair of 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.

In the scavenger hunt, 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 returned the vertex such that the distance described about was minimal. 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 overcrowded vertices.

SCAVY takes as input the vertex that the user is currently at. It then explores the network starting from that vertex using Dijkstra'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. In Figure 3 the small circles indicate intermediary vertices and the large circles indicate exit vertices. The arrows indicate edges in the network model and represent the hallways of the building. Intermediary vertices are used in computing routes through the building, but they are not valid end destinations. Each vertex's weight is maintained by the system so as to reflect how crowded it is; the weight of each vertex adjacent to a user's location and the weight of each exit vertex are considered in calculating the evacuation route. 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.

SCAVY, 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. 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.

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. Using an avatar allows us not to record personally identifiable information and eliminates any privacy concerns from the demonstration. We will bring about 100 of these cards for attendees. The system can be interacted with by using only the RFID front-end, the QR web front-end, or a combination of the two.

For demonstration of the RFID front-end, 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 front-end 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 front-ends will both access the Internet to communicate with our back-end database system. The RFID scanners are connected to machines which will access an available Internet connection. Since the QR-Code front-end 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 efficacy 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 that it can be used as long as desired.

5. CONCLUSION

In this work we have described a system designed to test the applicability of new technology to the problem of building evacuation. The goal has been to direct people to safe exits in a speedy and balanced manner. To achieve this we implemented a scavenger hunt game with these technologies – RFID and mobile wireless access – that was driven by the same principles as the proposed evacuation system. Our scavenger hunt system can be directly applied to many other indoor navigation settings, such as museum tours.

6. 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*, pages 231–232. ACM, 2007.

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

[4] Google Inc. Zxing. http://code.google.com/p/zxing/.

[5] B. Hoppe and E. Tardos. Polynomial time algorithms for some evacuation problems. In *SODA '94*, pages 433–441. Society for Industrial and Applied Mathematics, 1994.

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

[7] F. Lyardet, D. W. Szeto, and E. Aitenbichler. Context-aware indoor navigation. In *Conf. on Ambient Intelligence*, pages 290–307. Springer, 2008.

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