A Web Solution to Waiting Line Problem

José Torres Carreras
Computer Science
Juan M. Ramírez, Ph.D.
Electrical & Computer Engineering and Computer Science
Polytechnic University of Puerto Rico

***Abstract** – In this article I present an approach to solve the problem of wasting time waiting in a line for product or service. A solution is presented composed of a platform API and a mobile application. The solution provides a way to virtualize a line and also give insides on how to improve the system using queuing theory and the data collected in the platform.*

***Key Terms** – Mobile Applications, Service Oriented Architecture, Event Driven Architecture, Queuing Theory.*

INTRODUCTION

On January 29 2007 a new era on mobile computing was born with the release of the Apple IPhone. This fantastic device introduces features that can be exploited not only by the business but also for individuals in the form of small applications. Evidence of this extraordinary success is the actual number of applications already available for this device.[1][2] According to the apple official site there are over 500,000 of them and the number increase quickly every year. Many companies are also capitalizing on this success coping Apple business model. This is the case of Samsung, Motorola, Blackberry, HTC and many others.[3] The objective of this article is not promote a single company or a single device, instead is to present the huge opportunity the mobile computing presents to small companies and also present a new idea for niche app, one that can potentially help many people and save lot of waste time.[4][5] I am exploring the idea of using the already available technology embedded on smartphones to build a platform to follow a waiting line. The kind of lines many people take to get a service, or a product. To present the potential of this new app it is adapted to health care providers and medical offices.

SMARTPHONES TECHNOLOGY

Although the mobile phone has been widely used for several decades, smartphones are a more recent advance. They are mobile phones that offer not only the standard facilities such as voice and text communication, but also advanced computing and communication capability, including, for example, Internet access (always connected) and geo-positioning systems.[6] In comparison to earlier mobile phones (pre IPhone), smartphones generally also have larger, higher resolution display screens. Most of the newer generation of smartphones also incorporates other features such as on-board personal management tools, high quality cameras and recording devices. Specifically screen size is increasing not only on size but also on resolution. Some smartphones, such as the Blackberry, also incorporate small internal keyboards in their designs. Recently, Apple's iPhone and Google's Android touch screen devices have increased smartphone ownership. They are popular because of their intuitive and tactile graphical user interfaces and natural gesture control. The latest generation of smartphones is increasingly viewed as handheld computers rather than as phones, due to their powerful on-board computing capability, capacious memories, large screens and open operating systems that encourage application development.[7][8] The potential for the creation of simple and easy to download apps for smartphones has created a vibrant new industry. There is now an app for just about every social, entertainment and educational requirement.

Smartphones have now achieved such a

pervasive presence in society that users find it easy to self-organize themselves across large geographical areas. Many have adopted a culture where they are ‘always connected to their peer groups, communities of practice and information. The mobile phone provides an essential ‘any time, any place’ portal into the entire world wide web of knowledge. Such continuous and pervasive social connectivity has important implications for society, and holds a lot of potential in particular for use in education, healthcare and medicine.

PROBLEM

The initial idea for this mobile application came for personal experience during many visits to medical offices and my 15 years of experience working for a healthcare payer. The problem is this, many Medical Doctors in their practice schedule patients for a given date. There are doctors where patient amount per day is in excess of 50. For people with lots of time to waste this is not an issue but normally but not my case and for sure not the case of many people. Medical offices are not the only place where this situation happens. For example during enrolling period universities also have long lines of students waiting to be served. The same occurs at government offices, restaurants, business offices to mention only a few examples. This issue is recognized by the government of Puerto Rico when it states in law number194 (August 25, 200) and its amendment in 2008 that is not allowed for a patient to wait more than an hour to be attended in any medical practice. Although this can alleviate the law is not enforced and patients still are waiting long hours. Also this law is only for patients and as I mentioned on previous examples universities, business and other has the same issue. So how this can be solved?

SOLUTION

The waiting line problem can be divided in parts. First one needs to get a turn and second we need to monitor the line in some way. This is very obvious, what is the first thing you do when arriving to a line, get a turn and what is the second thing, monitor the line maybe asking to the dispatcher about the line status. The intrinsic problem is that both actions requires that a person be present all the time in order conserve the turn. With this in mind a solution can be to virtualize the turns and build a monitoring system that can be accessed from any smartphone. That way we can go to the line take a virtual turn and the go to do other thing while the line dispatch all turns before yours. From the smartphone we can monitor the line and go to get the service or product just in time. The problem of been at the service line location is not necessary. This solution can’t be reduced to a single smartphone application. A complete platform needs to be built to accommodate different services and characteristics of the physical waiting line. The purpose of this article is to present the concept of this service virtualization and for this reason specific details are beyond scope. .

ARCHITECTURE

Putting all components together requires some kind of guidance and for this reason I selected three well know patters for this solution. The architecture patters are:

- Service Oriented Architecture (SOA).
- Event Driven Architecture.(EDA)
- Enterprise Service Bus (ESB).

SOA is an architectural concept in which all functions, or services, are defined using a description language and where their interfaces are discoverable over a network. The interface is defined in a neutral manner that is independent of the hardware platform, the operating system, and the programming language in which the service is implemented. This facilitates and eventually migration to cloud services.

One of the most important advantages of a SOA is the ability to get away from an isolationist practice in software development, where each department builds its own system without any knowledge of what has already been done by others in the organization. This "silo" approach leads to inefficient and costly situations where the same functionality is developed, deployed and maintained multiple times. A SOA is based on a service portfolio shared across the organization and it provides a way to efficiently reuse and integrate existing assets, as shown in Figure 1.

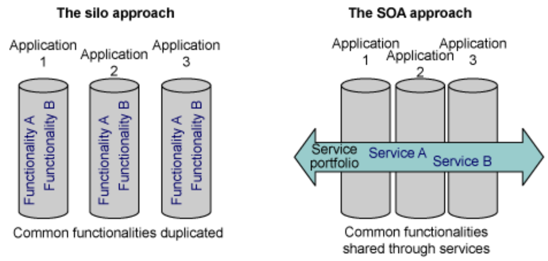


Figure 1
The "silo" Approach vs the SOA Approach

SOA is based on a conventional request/reply mechanism, as seen in Figure 2. A service consumer invokes a service provider through the network and has to wait until the completion of the operation on the provider side.

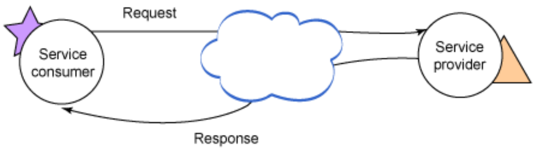


Figure 2
The Request/reply Mechanism in a SOA

In 2003, Gartner introduced a new terminology to describe a design paradigm based on events: Event-Driven Architecture (EDA). EDA defines a methodology for designing and implementing applications and systems in which events transmit between decoupled software components and services. EDA does not replace, but rather, complements the SOA. While SOA is generally a better fit for a request/response exchange, EDA introduces long-running asynchronous process capabilities. Moreover, an EDA node posts events and does not depend on the availability of a published service. It is really decoupled from the other nodes. EDA is sometimes also referred to as "event-driven SOA".

EDA uses messaging to communicate among two or more application processes. The communication is initiated by an "event". This trigger typically corresponds to some business occurrence. Any subscribers to that event are then notified and thus activated, as shown in Figure 3:

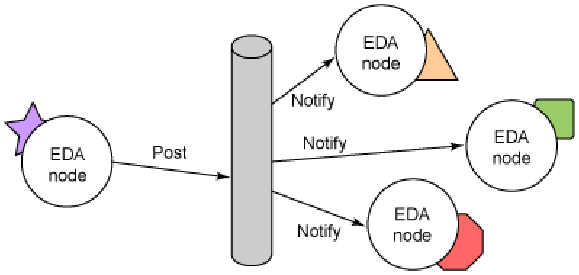


Figure 3
The Publish/subscribe Mechanism in an Event-Driven Architecture

An Enterprise Service Bus (ESB) combines event-driven and service oriented approaches to simplify integration of business units, bridging heterogeneous platforms and environments. The ESB acts as an intermediary layer to enable communication between different application processes. A service deployed onto an Enterprise Service Bus can be triggered by a consumer or an event. It supports synchronous and asynchronous, facilitating interactions between one or many stakeholders (one-to-one or many-to-many communications). So the ESB provides all the capabilities of both SOA and EDA paradigms.

An Enterprise Service Bus is an architectural pattern and can be implemented by many different products within the organization, and assembled together to act as a federated bus. More and more vendors are now offering a complete product to fulfill enterprise integration needs.

COMPONENTS

In order to have a basic understanding this list shows the main components of the solution. The components will be implemented as services for line providers and line consumers fallowing the SOA Architecture. Line providers are those that serves and administer a line, for the purpose of this article they refer to the medical office staff and line consumers that are the people that want to be served:

- Security Framework Services --the authentication module should provide a way to manage users, groups of users, and roles.
- Error Framework Services-- The platform needs to provide a standard way to indicate errors, as well as a way to specify how the application should deal with these errors (e.g. renders a special error page). In addition, it would be great for this to be tied somehow with a bug/issue tracking system, so that application error can be automatically logged to the developers
- Logging Framework Services -- Since all the user-visible output goes to a client browser, there needs to be way that the application can trace its behavior. In the available platform, there should be a simple mechanism for logging.
- Data Store Services –.A way to store information that is shared and synchronized between the various threads is absolutely necessary for the higher levels of the application framework. It would also be used by other components, such as user authentication. It makes it easier to integrate the various components if the concrete implementation allows sharing the database connection or cursors with other components, so that we can minimize the amount of resources that the entire framework uses for each thread.
- Internationalization Services – The need to support rendering pages in different languages requires a bit of preparation and some support libraries.
- Notifications Services – This is one of the core components of the solution. It is use to respond to real-time changes and events as they occur. In the following sections this component will be explained.
- Application Services – this conforms all the functionality and capability offered by the waiting line virtualization platforms. It includes many services that will be enumerated at a latter section.

The services will be packed in the form of an Application Program Interface API and maybe distributed as a System Development Kit (SDK) to facilitate the adoption of third party software. Some services offered by the waiting line virtualization platform when subscribed as a line provider could be:

- Create a virtual Line.
- Accept subscriptions to specific virtual line.
- Remove subscriptions from specific line.
- Remove a Virtual Line.
- Calculate waiting time.
- Give information about waiting line size
- Send notifications to subscribed people. This can be as broadcast or specific. For example broadcast a message with notifying to have everything ready at the time to be served.
- Rise events to specific situations.
- Split a Virtual Line. This is useful in situations where the line is growing fast and a dispatcher is added.
- Join two Virtual Lines.

The same SDK will be used to develop the mobile app.

MOBILE APPLICATION

The previous section discussed the server part of the solutions and now is time to discuss some characteristics of the mobile app. The application is oriented towards the healthcare business but because an SDK will be used adapting it to new business like a university will be easy. The first thing I'm trying to accomplish is to complement and in some cases substitute completely the ID card health insurance companies give to its members.[9][10[11] The common scenario for healthcare payers or healthcare insurance companies is that a single card with basic information is given to all its members to help healthcare providers know how to bill for services.[12][13][14] The ID could include:

- Your name
- Your Priority Health contract number
- Your group number (if you're a member of an employer's health plan)
- Provider network information
- Dependents (spouse, children) covered by your plan
- Priority Health's billing address
- Priority Health Customer Service numbers
- What network to call when you're traveling to other parts of the U.S.

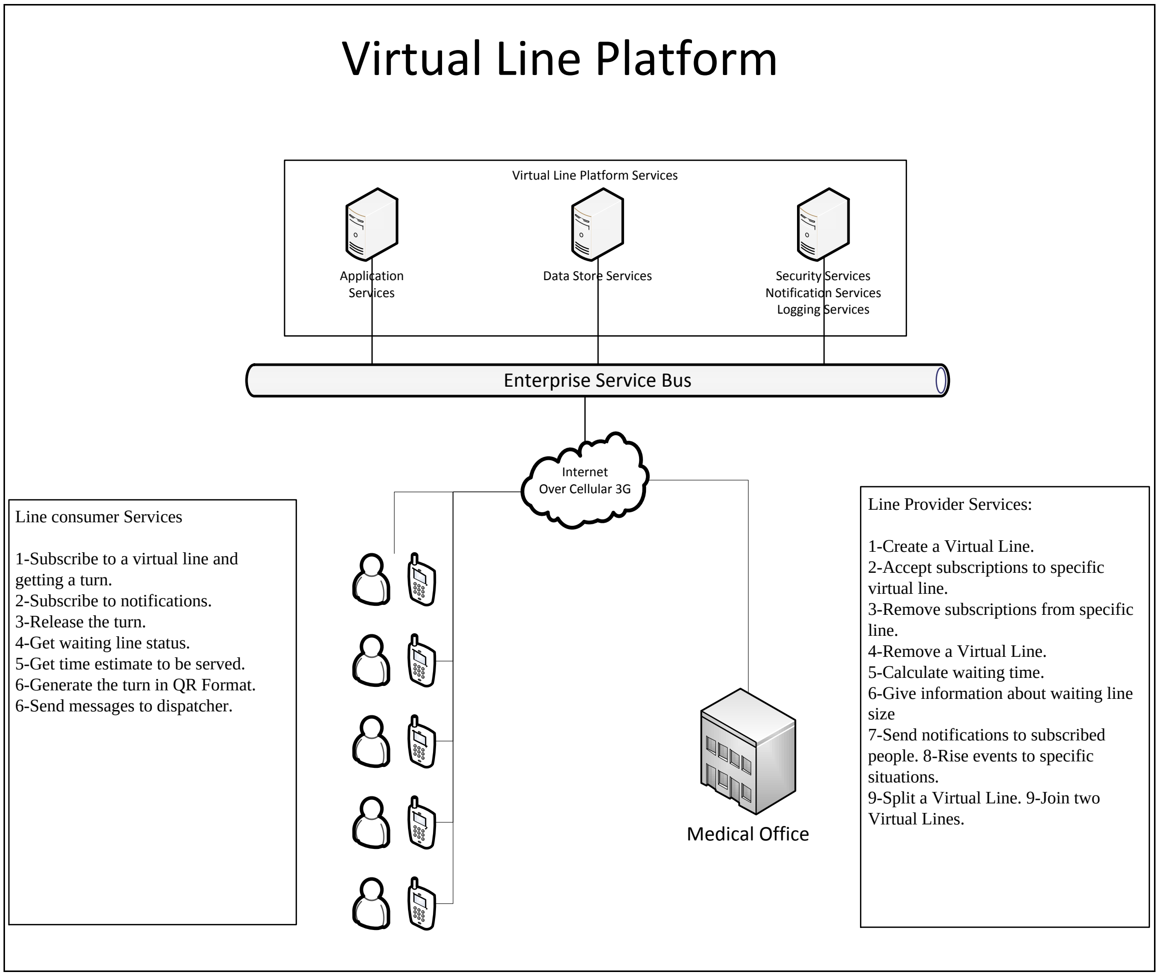
The problem with this approach is that the card is static media.[15][16][17][18] Once it is printed the information is permanent. Some consequences are that it can be lost, information can be wrong; benefits can change and is prone to fraud. Maybe you are asking the relation with this article? Well, in this article, I'm proposing a mobile app to virtualize a waiting line, specifically the ones at medical offices. The way I conceptualize the app is that it can pull the card information from the health insurance web site and store this information in the mobile permanent storage.[19] This information will be used to identify yourself at the physical line location and get a turn or ticket. The information exchange is through services provided by the waiting line platform so errors in typing are eliminated.[20] Now that the app has some information then it needs some features. This are some of the services offered by the waiting line virtualization platform when subscribed as a line consumer:

- Subscribe to a virtual line and getting a turn.
- Subscribe to notifications.
- Release the turn.
- Get waiting line status.
- Get time estimate to be served.
- Generate the turn in QR Format. It will be useful for scanning directly from the form if needed.[21]
- Send messages to dispatcher.

These features are enough to solve the waiting line problem and give people the freedom of doing whatever they want while the line is performing. The complete system is presented on figure 4.

BEYOND A WAITING LINE

The services provided by the server on this solutions attempt to free people from waiting lines but is doing nothing to help improve the service time. Although is out of scope the virtual line platform includes a logging mechanism. The information persisted on logs could be used to analyze the behavior of the line with the help of mathematics. Queuing theory is the mathematical study of waiting lines, or queues. The theory enables mathematical analysis of several related processes, including arriving at the (back of the) queue, waiting in the queue (essentially a storage process), and being served at the front of the queue. The theory permits the derivation and calculation of several performance measures including the average waiting time in the queue or the system, the expected number waiting or receiving service, and the probability of encountering the system in certain states, such as empty, full, having an available server or having to wait a certain time to be served. This information can give critical information to improve the system. Maybe it is too complex for a medical office with an average of 50 patients, but is critical to server large waiting lines at government offices that serve a large group of people.



Line consumer Services

1-Subscribe to a virtual line and getting a turn.
2-Subscribe to notifications.
3-Release the turn.
4-Get waiting line status.
5-Get time estimate to be served.
6-Generate the turn in QR Format.
6-Send messages to dispatcher.

Line Provider Services:

1-Create a Virtual Line.
2-Accept subscriptions to specific virtual line.
3-Remove subscriptions from specific line.
4-Remove a Virtual Line.
5-Calculate waiting time.
6-Give information about waiting line size
7-Send notifications to subscribed people. 8-Rise events to specific situations.
9-Split a Virtual Line. 9-Join two Virtual Lines.

Figure 4
Complete Virtual Line Solution

REFERENCES

[1] MD Smartphone Adoption Growing Exponentially. (2010). *Review of Ophthalmology*, *17*(9), 10-12.

[2] Latest Trends in Mobile Handset Design. (2010). *Microwave Journal*, *53*(6), 46.

[3] Boulos, M., Wheeler, S., Tavares, C., & Jones, R. (2011). How smartphones are changing the face of mobile and participatory healthcare: an overview, with example from eCAALYX. *Biomedical Engineering Online*, *10*(1), 24-37. doi:10.1186/1475-925X-10-24

[4] Tolliver-Nigro, H. (2011). DSCOOP Trends: Mobile and Social. *Seybold Report: Analyzing Publishing Technologies*, *11*(4), 10-12.

[5] Rodríguez, J. C., Oliveira, M., & Vaidya, B. (2010). New Trends on Ubiquitous Mobile Multimedia Applications. *EURASIP Journal On*

[6] Chan, A. S., Cao, J., Chan, H., & Young, G. (2001). A Web-Enabled FRAMEWORK for SMART CARD Application in Health Services. *Communications Of The ACM*, *44*(9), 76-82.

[7] Mair, A. (2009). Big numbers and small screens: Challenges and opportunities of the mobile web for national statistical offices. *Statistical Journal Of The IAOS*, *26*(1/2), 47-54. doi:10.3233/SJI-2009-0701

[8] *Wireless Communications & Networking*, 1-11. doi:10.1155/2010/689517

[9] Moss, B. (2002). Do biometrics invite a lackadaisical mindset?. *Health Facilities Management*, *15*(9), 22.

[10] Preferred Health introduces new Health & banking card combo. (2003). *Caribbean Business*, *31*(29), 51.

[11] Hodgson, K. (2011). The Future is in the Card (Readers). *SDM: Security Distributing & Marketing*, *41*(10), 78-82.

[12] Ni, A., & Tat-Kei Ho, A. (2008). A Quiet Revolution or a Flashy Blip? The Real ID Act and U.S. National Identification System Reform. *Public Administration Review*, *68*(6), 1063-1078. doi:10.1111/j.1540-6210.2008.00955.x

[13] Shelfer, K. M., & Procaccino, J. (2002). SMART CARD EVOLUTION. *Communications Of The ACM*, *45*(7), 83-88.

[14] M'CHIRGUI, Z. (2010). SMALL WORLD CHARACTERISTICS OF INNOVATIVE SMART CARD NETWORKS. *International Journal Of Innovation Management*, *14*(2), 221-252. doi:10.1142/S1363919610002623

[15] Sullivan, R. J. (2008). Can Smart Cards Reduce Payments Fraud and Identity Theft?. *Economic Review (01612387)*, *93*(3), 34-62.

[16] Cowell, F. A. (2008). Problems and Promise of Smart Cards in Taxation. *National Tax Journal*, *61*(4), 865-882.

[17] MGMA Launches Project SwipeIT. (2009). *Health Care Collector: The Monthly Newsletter for Health Care Collectors*, *22*(10), 11-12.

[18] Wisz, M. (2009). The Eyes Have It. *Health Management Technology*, *30*(3), 34-35.

[19] LONG, B. (2010). ER must-haves: ID, insurance card…and reading glasses. *Employee Benefit News*, *24*(7), 14-16.

[20] Martin, N. J., & Rice, J. L. (2010). Building better government IT: understanding community beliefs and attitudes toward smart card technologies. *Behaviour & Information Technology*, *29*(4), 433-444. doi:10.1080/01449291003767938.

[21] Ferrer-Roca, O. O., Marcano, F. F., & Diaz-Cardama, A. A. (2008). Quality labels for e-health. *IET Communications*, *2*(2), 202-207. doi:10.1049/iet-com:20060596.