A Service-Oriented Architecture for Optimal Service Selection and Positioning in Extremely Large Crowds

Mohammad A.R. Abdeen
The Faculty of Computer and Information Systems
The Islamic University of Madinah, Madinah, Saudi Arabia

***Abstract*—The problem of managing large crowds has many aspects and has been reported in the literature. One of these aspects is the distribution of supplies such as food and water in especially when the targeted region is overcrowded. Some of the challenges is to plan the locations of food and water supply centres in such a way to achieve multiple objective functions such as the type of food and the shortest distance to customer. A practical example of this problem is the food distribution and food cart location in the region of Mena (also known as Tent City, in Saudi Arabia) during the yearly pilgrimage season. In this work, we propose a Service Oriented Architecture (SOA) for positioning services in the region of Mena (Mecca – Saudi Arabia) that covers an area of approximately 20 square kilometres during the pilgrimage season. The architecture proposes an optimal service selection as well as a mobile food cart positioning algorithm based on client pre-set profiles to achieve multiple objective functions for the clients as well as the service providers. Some of these objective functions are the least waiting time to be served, the shortest distance to service, the lowest cost, and the maximum profit for the service provider.**

Keywords—Large crowd management; Service-Oriented Architecture; multi-objective optimization; Hajj; Mena; WSDL

I. Introduction

Each year, millions of pilgrims travel to Mecca from various places/countries all over the world to perform their once-in-a-life-time duty of pilgrimage (Hajj). The trip lasts for at least five days in which pilgrims move from a place to another within an area of approximately 50 square kilometers (12x4 Km). Managing such a large number of people, that reaches two million, is a significant challenge to the event organizers. A variety of services are offered to those pilgrims including, the supply of food and water, accommodations, healthcare, guidance, and many others. Facilitating such services for such a large number of people and in such a relatively small area with dense population is a major undertaking. One of the regions that pilgrims pass through and stay for three days is called the region of Mena, a.k.a. the “Tent City”. This place is located in the open desert and it is economically infeasible to build infrastructure to house pilgrims and services for just three days in the whole year. The pilgrimage’s organizers have resorted to the idea of pre-prepared tents that are easily installed and moved around. Fig. 1 shows the region of Mena [1].

With today’s technology and the vast availability of mobile devices, there comes an opportunity to organize such events with better efficiency and convenience for both the pilgrims and the organizers.

Food and water distribution in the region of Mena (in Mecca) is usually done by trucks carrying various food supplies [2]. Also, the authorities have recently depended on water containers for water distribution in the region of Muzdalifa (few kilometres away from Mena) [3]. A dire question is where to locate those food mobile trucks and water supply containers so that an optimal service is provided to minimize the effort and time for pilgrims.

This work proposes a Service Oriented Architecture (SOA) that facilitates positioning mobile service carts for the purpose of mutual benefit for both the consumers (pilgrims) and the service providers (such as food, hairdressing, telephone services, etc.). Since the area of Mena has some permanent but insufficient installations for such service provisioning, this architecture proposes the smart use of mobile service carts to provide the required services. The architecture uses the GPS location of clients in addition to their pre-set profile in which they specify the type of service required. The system proposes to clients the most convenient service (with shortest distance, and least service times). The architecture also uses the customer information to dynamically locate mobile carts to achieve the same goals based on clustering criteria that combines the customer’s and the service provider’s own criteria.

This paper is organized as follows: Section 2 presents a background of the topic and the definition of the main technical terms. Section 3 presents the proposed SOA. Section 4 shown the implemented prototype of the system. Conclusions and future work are presented in Section 5.

Fig. 1. The Region of Mena – Mecca (The Tent City).

II. Background

Large crowd management has been addressed in several works in the literature [4][5][6].

In [4], the authors presented a broad review on various crowd management and monitoring technologies that are based on vision (CCTV), Wireless/RF and Web/Social media technologies. It presented two of the most known examples of the crowd management, viz., the Hajj (pilgrimage) and the Kumbh Mela (in India). They addressed the events form the security view point and to employ the latest technology to avert disastrous situations.

The authors in [5] presented a Near Field Communication (NFC) based architecture for providing various services in large crowd situations. The architecture relied on mobile phones that are NFC enabled. The authors presented the Hajj (pilgrimage) as a case study. They considered services such as medical emergencies, pilgrim identification, and lost pilgrim help. They implemented a proof-of-concept on a Samsung smart phone that runs Google Android OS. A Microsoft SQL server is used in the backend.

In [6], the authors presented an architecture for providing a specific, yet important, service for individuals in large crowds. They presented an in-memory architecture for locating, tracking, and guiding astray pilgrims during Hajj season in Mecca. The simulation results showed that the architecture scales well with the number of pilgrims in the system.

In [7] the authors presented an architecture to manage large crowds in the region of Mecca during the pilgrimage (Hajj) season. The architecture addressed managing the large crowds including security issues and using RFID to track pilgrims. However, they did not take into account the provision of food and other services.

A. *Servie Oriented Architecture*

Service oriented architectures is a software approach to developing efficient general-purpose distributed applications from specialized components called services. This approach significantly improves productivity and reusability. The SOA has many useful applications in the banking [8], healthcare [9], mobile-learning (m-learning) [10], automotive [11] and many other sectors.

Services are the central part of the architecture which are components with well-defined interface which are raceable from any location in the network. Each service defines a contract with which the service is accessed. The architecture has essentially three components, the service provider, the service requester and the service registry. Fig. 2 shows the components of a SOA.

Some standards are an integral part of the SOA, such as the WSDL, the UDDI, and the SOAP. In the following sections these topics are briefly covered.

B. *Web-Service Description Lanuage*

Services need to identify themselves to the service registry. The Web Service Description Language (WSDL) is an xml-based language that enables service providers to describe their offered service in a machine-readable format. It includes definitions of how the service is called, what parameters it takes, and what data structures it returns. A WSDL file is an XML-based file and is where the service description is stored. It has several components including the service definition, the data types, the messages, the port type, the binding, the ports, and the services. The service definition in the form of a WSDL file is published to a service repository [12].

C. *Universal Discovery and Description Integration*

An important component of the SOA is the service registry. This component provides for the ability to search and discover the registered services. Service requesters are the primary customer for such a component. The Universal Discovery and Description Integration (UDDI) is a technology based on XML markup language and was developed by a consortium of more than 300 businesses and technology leaders to enable companies and/or applications to quickly find web services. The UDDI registry is essentially a directory and consists of three sub-directories:

- The white-pages: where organizations and service providers list their names, phone numbers, and address.
- The yellow-pages: where more detailed information such as business classification according to the north-American standard.
- The green-pages: where information about the business process is included such as shipping, billing, and purchasing methods.

D. *The Simple Object Access Protocol*

A standard protocol is needed to wrap all messages that are being exchanged between the components of the SOA such as messages between the service provider and the registry service, the service requester and the registry service and finally the service requester and the service provider. The Simple Object Access Protocol (SOAP) just provides that. It is a standard XML-based wrapper protocol that wraps all communication for a SOA application. A typical SOAP message consists of an envelope that contains a header and a message body. The header contains information on how the message is to be processed including routing and authentication information. The message body, however, contains the actual message to be processed and delivered [13].

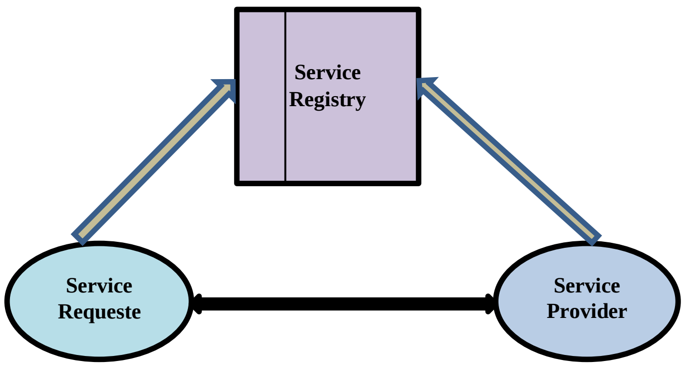


Fig. 2. The Service-Oriented Architecture.

(*sponsors*).

III. The SOA Architecture

In this section we describe the overall high-level SOA architecture for the proposed system.

The presented architecture consists of three main components similar to the standard SOA, as shown in Fig. 3. However, the type of component and its behavior is what enables this architecture to serve its purpose. The service requester is the client requesting various services, such as food services, telephone service, hairdressing service, transportation services. The service provider is the entity that provides the variety services for pilgrims and visitors as previously mentioned. The registry is the entity where all services have to register with prior to being acknowledges in the system and to enable the search and discovery by the client.

A. *The Clients (Service Requester)*

The client is the service requester and it runs a mobile application on a mobile device. Initially the client selects through the mobile App the specific service from a radio-button selection menu. The client also selects his/her preferences such as type of food for example (ethnic food, fast food, Veggie) or mobile service provider preferred (Vodaphone, Bell, Orange Telecom). This data is sent to the service registry for recommendations.

B. *The Registry*

The registry is an entity (a process) that runs on a separate machine that keeps information about the services that are registered with it. In this architecture, the registry keeps information about the food service providers, the phone service providers, and other service providers. A service providers registers to the registry by sending its information in a WSDL message. The registration information includes, name of service provider, type of service offered, return value type (a data structure in this case).

C. *The Services*

There are various types of services supported by the system, the food services, the mobile phone service, the hair dressing service, and the transportation service. These services register initially to the service registry and provide information regarding the provided service. This information includes the following items:

- Service Type (Food, Hairdressing, telephone, …)
- Service GPS Location
- Service cost.
- Available Deals/discounts
- Average time to serve
- Delivery available (Yes/No)

Service providers are in the form of mobile carts and are initially located as per a “first guess” provided by the service optimizer component. The service providers move later to a possible future location as per the recommendation of the optimizer which utilizes an algorithm based on centroid calculations.

D. *The Service Optimizer*

In this architecture an extra service is proposed that has a central role for the system operation. This service is provided by the facilitator or the framework. It receives location information from service requester as well as the optimization criteria as per service requester (whether the requester wants to optimize cost, delivery time, or distance) and it replies with the service provider reference that achieves the requested optimization criteria.

Fig. 4 depicts the sequence diagram of the proposed architecture and the interaction of its components.

As shown in the sequence diagram, the client selects the type of service from a mobile App on the client smart phone. The client is provided with further detailed choices according to the type of service selected. As an example, if the client selects food service, a list of other options such as type of food (ethnic, fast food, special diet – Veggie, Gluten free) is displayed. Other options such as pick-up or delivery options are provided. This request is forwarded to the service registry which keeps a list of all registered service providers. Supported service providers are required to register their service with the registry and provide service details such as service type, service location, average waiting time, and whether or not there is a delivery option. Upon receiving a client service request, the registry service replies back with a list of possible candidates that provide the requested service. The client then retrieves the references of the received candidates and calls the optimizer service after obtaining its reference and sends the records of the service candidates.

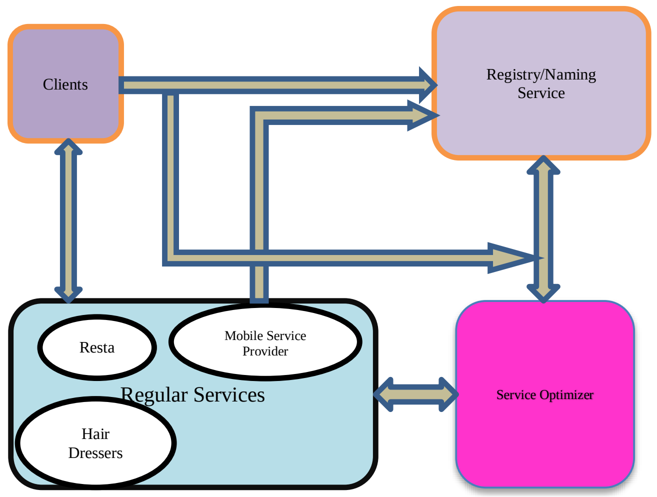


Fig. 3. The Proposed SOA for Food and Service Distribution in the Region of Mena, in Mecca During Pilgrimage.

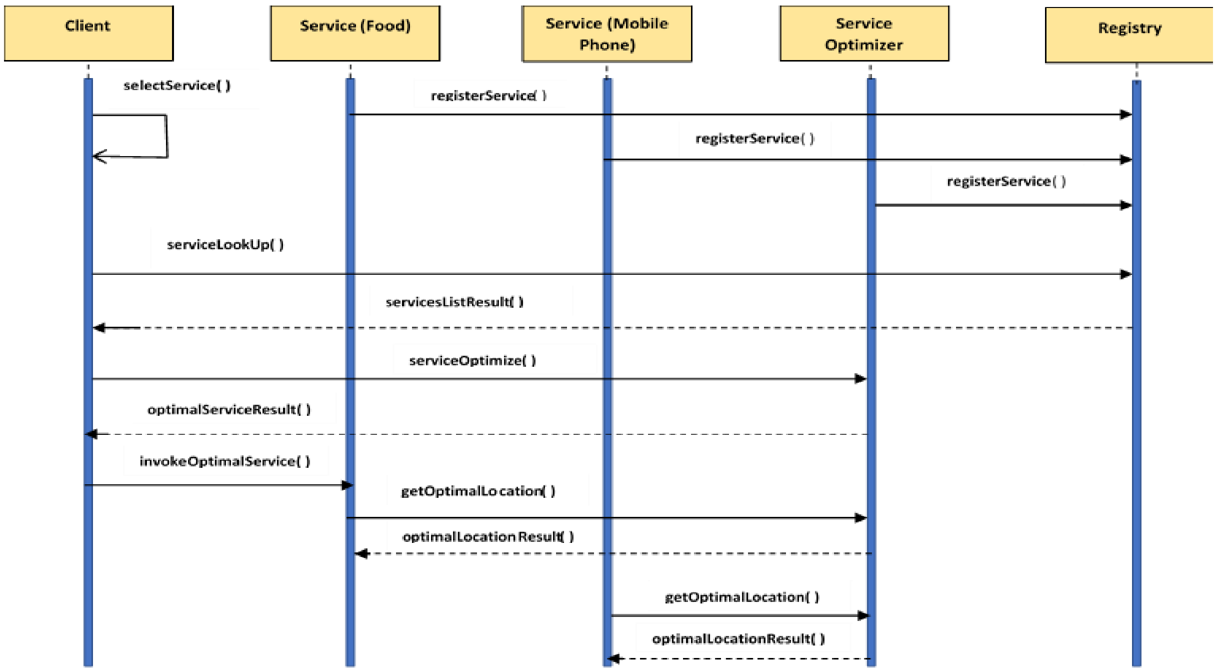


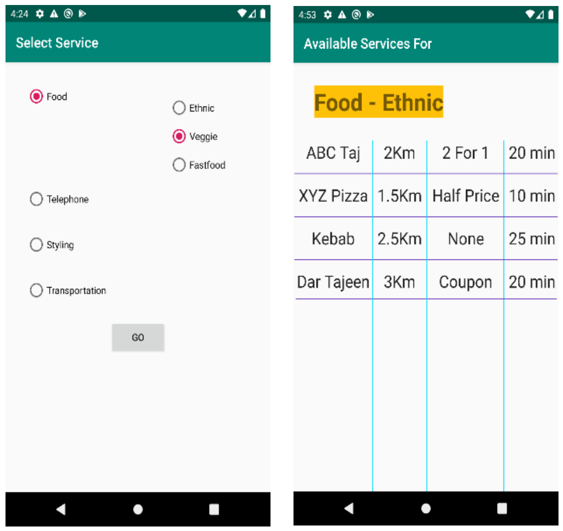
Fig. 4. The Service Distribution Sequence Diagram.

1) Thin client–This architecture suits the utilization of thin clients where only data presentation is handled by the client and the actual computation and optimization algorithms are implemented on the server side. This architecture suits well the majority of the mobile devices and to maintain the availability of this type of service on as many devices as possible to suit the diversity of the pilgrims and their mobile devices of choice.

2) Fat client–The architecture can also support fat client in which client device handles both the data presentation as well as the data processing algorithms. In this case, the client device downloads the mobile App as well as a plugin that implements the optimization algorithms. In this situation, the client needs not to communicate back-and-forth with the service optimizer component.

IV. Prototype Implementation

We have implemented a prototype of the proposed architecture using Android Studio [14] as the mobile App development environment and the Visual Studio [15] environment. Fig. 5 shows the initial menus of the developed mobile App. The first menu shows the menu items at the client side which lists the available service to the client such as food, styling, telephone, and transportation. If a client, for example, selects the food option, a submenu appears showing the subitems (such as Ethnic, Veggie, and Fast food). Upon the client pressing the "Go" button, a SOAP message is sent to the registry service requesting the availability of the selected item. The registry responds with a reply SOAP message showing all options in the required class and the reference to every service. These services have previously been registered to the registry with an WSDL message. The client uses the reference information to directly contact the service providers to obtain their relevant information (as shown in the right graph of Fig. 5).



[✓ Food]
Ethnic
[✓ Veggie]
Fastfood
Telephone
Styling
Transportation

Food - Ethnic

ABC Taj	2Km	2 For 1	20 min
XYZ Pizza	1.5Km	Half Price	10 min
Kebab	2.5Km	None	25 min
Dar Tajeen	3Km	Coupon	20 min

Fig. 5. The Mobile App Showing the Service Selection and Detailed Service Response.

Clients then can opt to optimize the list of services provided by the service providers. Clients can call the Optimze_Service() service and provide their optimality criteria. This service sends back the response to the client with the optimal list. The code Listing 1 shows the Optimize_Service() web service.

On another hand, this system provides an optimal location service to service providers. In other words, a service provider can request that it sent the location where the majority of customers reside. As an example, if a given food cart offers ethnics food of a specific ethnicity (e.g. Indian) then the service provider can request to be sent the optimal location that provides the minimum average distance to the majority of those clients. Fig. 6 shows a schematic of the optimal position calculation based on the concept of centroid. If the clients are scattered across a given area, then the optimal location of a service cart is the centroid of the service requester as shown by Eqn. (1).

$$Centroid = \frac{1}{n}\sum_{i=1}^{n} d_i \qquad (1)$$

where d_i is the actual distance between an initial centroid location and a given client, and n is the number of clients belonging to the class of service (such as ethnic food). Listing 2 shows the code for **Optimize_Location()** web service that is used by the service providers.

```
enum OptimalVals {proximity,
   waitingTime, Price, Deal};
class GPS{int x; int y};
class Criteria{
   GPS clientGPS;
   OptimalVals[ ] List = new
OptiamlVals[5];
}
class cItem{
   string name;
   float  dist;
   float  discount;
   int    waitM; //waiting time in
minutes
}

//this function is a service called by
//a service requester to obtain best
// service according to a priority
// list he/she provides.
cItem Optimize_Service(Criteria
OptCria){
         Console.WriteLine("Optimizing
                Service");
        cItem CI = new cItem();
        return CI;
}
```

Listing 1 Sample Code for the Service Optimization Web Service.

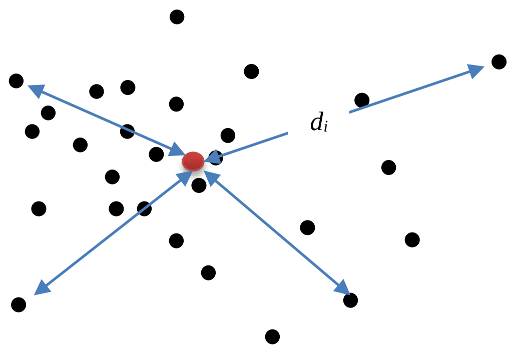


Fig. 6. The Client Centroid Calculation.

```
enum ServiceTypes {Food, Styling,
Telephone, Transportation};

enum ServiceSubTypes{ Ethnic, Veggie,
Fastfood, HairDress, Vodaphone, Orange,
STC, Taxi, Bike, Bus};

class ServiceType{

   ServiceTypes type;

   ServiceSubTypes  stype;

}

GPS Optimize_Location(ServiceType sT){

   Console.WriteLine("Optimizing
Location");

   return gps;

}
```

Listing 2 Showing the Optimize_Location(..) Web Service.

V. Conclusions and Future Work

In this paper we presented an SOA architecture for service provisioning and optimal service location in events with extremely large crowds. We considered a case study of the Hajj (pilgrimage) and more specifically the region of Mena where fixed installations and facility are not economic justifiable and mobile service carts are more appropriate. A prototype of the architecture was implemented as a mobile App and services are implemented with Android Studio and Microsoft Visual Studio. One important component of the system is the service optimizer. Upon receiving a client request for a service and based on the client optimal criteria, the system returns a sorted list of the recommendations including the average waiting time to get the service. In addition, the services can solicit the optimal location from the service optimizer to be convenient for the majority of the service requester. Optimization of the service location includes commuting times, and proximity of a special-type customers (such as when selling ethnic food).

As a future work, scalability of this architecture is an important issue. The architecture should be able to support hundreds of thousands of users simultaneously. A service registry could become a bottle neck due to surges in requests during peak times. In addition, security is another concern. Since service are public, it is then available for productive and nonproductive users. It is of equal importance to prepare for hundreds of thousands of legitimate users as well as for other users that might misuse the service.

Acknowledgements

The author would like to thank Dr. Khaled Bashir for his help with the mobile App in this work.

References

[1] The International Business Times Available From https://www.ibtimes.co.uk/hajj-2014-photos-vast-crowds-over-two-million-muslim-pilgrims-gather-mecca-1468734 [last accessed: December 23, 2019].

[2] Al-Watan Newspaper , Available From https://www.alwatan.com.sa/article/67972/اقتصاد/يقدم-أكبر-مائدة-طعام-وشراب-لملايين-الحجاج-في-مدينة [Accessed: December 2019].

[3] Saudi Press Agency (SPA), Available From, https://www.spa.gov.sa/1957009 [Accessed: December 2019].

[4] D. Sharma, Amol P. Bhondekar, A. K. Shukla, and C. Ghanshyam. "A review on technological advancements in crowd management." Journal of Ambient Intelligence and Humanized Computing 9, no. 3 (2018): 485-495.

[5] M. A. Mohandes,. "Mobile technology for socio-religious events: a case study of NFC technology." IEEE Technology and Society Magazine 34, no. 1, 2015, pp. 73-79.

[6] M. A.R. Abdeen and A. Taleb. "A Real-Time Algorithm for Tracking Astray Pilgrim based on in-Memory Data Structures." International Journal of Computer Science and Applications 9, no. 10, 2018, pp. 467-474.

[7] M. Yamin and M.A. Albugami, “An architecture for improving Hajj management”, In: Liu K, Gulliver SR, Li W, Yu C (eds) Service science and knowledge innovation ICISO 2014 IFIP advances in information and communication technology, vol 426. Springer, Berlin, 2014.

[8] J. DiMare, and R. S. Ma. "Service oriented architecture Revolutionizing today's banking systems.", IBM Global Business Services, 2009, [Available From: https://www.coursehero.com/file/32455299/SOApdf].

[9] K. Avila P. Sanmartin, D. Jabba, and M. Jimeno. "Applications Based on Service-Oriented Architecture (SOA) in the Field of Home Healthcare." Sensors 17, no. 8 , 2017.

[10] M. J. C. Guerrero, , M. González, M. A. Forment, and F. J. García-Peñalvo. "Applications of Service Oriented Architecture for the Integration of LMS and m-Learning Applications." In WEBIST, 2009, pp. 54-59.

[11] M. Wagner, A. Meroth, and D. Zöbel. "Developing self-adaptive automotive systems." Design Automation for Embedded Systems 18, no. 3-4, 2014, pp. 199-221.

[12] Web Service Description Language Version 2.0, Available From https://www.w3.org/TR/2007/REC-wsdl20-20070626/ [Accessed: December, 23, 2019].

[13] S. Graham, G. Daniels, D. Davis, Y. Nakamura, S. Simeonov, P. Brittenham, P. Fremantle, D. Koenig, and C. Zentner, Building Web services with Java: making sense of XML, SOAP, WSDL, and UDDI, SAMS publishing, 2004.

[14] The Android Studio, Available From, https://developer.android.com/studio [Accessed: December 2019].

[15] The Microsoft Visual Studio Available From https://visualstudio.microsoft.com [Accessed: December 2019].