International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988, Volume 5 (2013) pp. 685-695

Generic Service Patterns and Navigational Access Structures for Web-Enabled Public Healthcare

Marina Nazih[1], Ghada Alaa[2]

[1] Faculty of Informatics and Computer Science
The British University in Egypt
Ismailia Desert Road, El Sherouk City, Cairo, Egypt
Eng_marina_2011@hotmail.com

[2] The Egyptian Cabinet's Information & Decision Support Center (IDSC)
1 Magles El Shaab Street, Cairo Egypt
ghada_alaa@yahoo.com

***Abstract*: Public healthcare systems are complex with diverse components that utilize different technologies, therefore Service Oriented Architecture (SOA) is suggested as a data bridge between the heterogeneous parts. This paper investigates an Egyptian public healthcare case study where SOA is used for integrating hospitals with each other and with other healthcare actors through a governmental Enterprise Service Bus (ESB). Interactions between public hospitals through a governmental portal are known as Government-to-Government (G2G) services. The research will focus on architectural and navigational design aspects of the ESB. The service patterns are the contribution of this paper and they represent generic healthcare activities invoked by five major actors; the government as facilitator, the producers to cover the doctors and nurses, the healthcare providers represented in hospitals and clinics, the health intermediaries to include health insurance companies and pharmacies and finally the patient. Re-usability, abstraction, loose coupling and scalability are the software metrics used to evaluate the architecture following the proposed patterns. It is also shown how system navigation has been improved with the use of patterns.**

Keywords: *Service Oriented Architecture (SOA), public healthcare, Government to Government (G2G), Web Navigation, Windows Communication Foundation (WCF)*

I. Introduction

Service Oriented Architecture (SOA) is an architectural methodology that helps integrate heterogeneous systems, as it provides a data bridge between incompatible technologies. A system based on SOA will package functionality as interoperable services that can be re-used by multiple systems from diverse domains [18].

Different challenges are raised in the information technology (IT) domain to meet the needs of real world businesses which started from the first part of this decade. These challenges are represented in the increased complexity of modern IT applications and the increased demand for evolution, maintainability and continuous upgrades to new technologies. The main reason of complexity lie in the diversity of programming paradigms that leads to heterogeneous systems. Service Orientation (SO) is one of the latest solutions to attack the complexity of today's heterogeneous IT environments and hide the details of implementation from business users [1]. Thus, the new architectural approach "SOA" implies the encapsulation of business functionalities in form of re-usable services. SOA will help businesses achieve better return on investment (ROI). This is realized by wrapping business functions with a separate service interface layer that is interoperable and will exist well beyond the lifetime of any system it represents [1].
Public health care is a governmental act to help societies protect and promote health, and limit the spread of illness, injury and disabilities [16]. Although healthcare professionals and patients require accurate record keeping and improved communication that are deemed necessary to ensure accuracy of diagnosis, in Egypt computers are still not central to healthcare mission, as currently information is conveyed from one healthcare professional to another through paper notes or personal communication. In this regard, SOA can provide a good solution for automating partial parts of the healthcare value chain where services will be incrementally added with time. This is possible as SOA separates the service interfaces from their implementations and thus ensures easy integration of new services within the business process. Because each service is exposed through a standard interface, the underlying implementation of the business activities can be changed without affecting how the service is consumed [6].

In this paper we aim to provide re-usable functional and architectural patterns that would improve the development of integrated public healthcare systems based on SOA concepts. These patterns will decrease the complexity of the system and will improve its performance as the system will be broken down into flexible, re-usable components.

We also aim to build navigational access structures following the proposed patterns that would improve usability of healthcare services and their outreach to the consumers. Ease of use and improved navigational access of healthcare services through the Web portal are important features that should be taken into consideration as healthcare users in Egypt do not have full experience and convenience with information technology. For example, doctors and nurses are used to deal manually with papers and will need to change their habits and acquire new skills to run the automated health services.

II. Background on SOA & Web Services

According to IBM [7], SOA is an IT architectural style that supports the transformation of a business into a set of interlinked services that can be accessed when needed over a network. It is an approach to loosely coupled, protocol independent, standards-based distributed computing. Loosely-coupled means that these services should be independent of each other so that changing one of them should not affect the other services roles. As shown in Figure 1 a service is consumed by a service consumer through sending a service request where the service provider responds to the request through a service response.

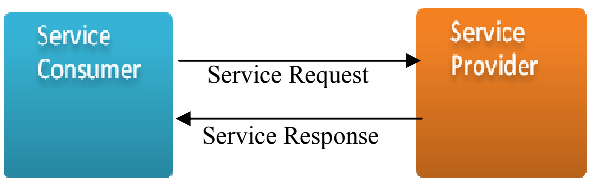


Figure 1. Basic Service Oriented Architecture (SOA)

According to Mahmoud [12], a web service is a software system designed to support interoperable machine-to machine interaction over a network. Web Service Based Systems (SBS) are essentially distributed in nature and in most cases are governed by third party service repositories. [10]. The provider of a web service publishes the service on UDDI (Universal Description Discovery and Integration) which represents the service broker (repository) so that the client can discover the service that suits him/her by searching the UDDI as shown in Figure 2. A web service is invoked by the client application (consumer) over the network where the service is hosted on the provider not the repository.

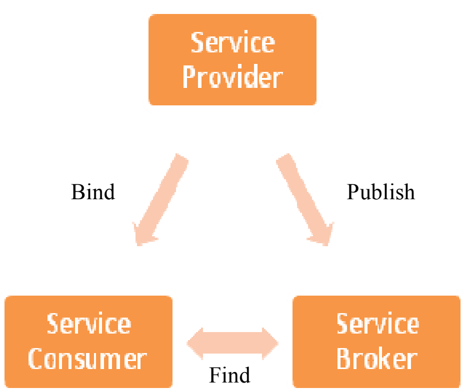


Figure 2. Web Service Operations

III. Web Service Technologies

Web services have been introduced as a cost-effective solution for integrating heterogeneous system components to overcome operating system, platform, and programming language barriers that were previously impassable. Web services are powered by XML and three other core technologies: WSDL, SOAP, and UDDI. [4]

A. Universal Description, Discovery and Integration (UDDI)

The UDDI allows the services to be listed over the Internet as it represents a registry mechanism for services. UDDI is an open industry initiative, sponsored by the Organization for the Advancement of Structured Information Standards (OASIS). Publishing and discovering services are made through the UDDI, as it defines how the web service can be invoked and it is platform-independent as the web service descriptions are represented as XML files.

B. Web Services Description Language (WSDL)

Web services functionalities are listed in form of XML format in the WSDL document. The WSDL includes method names, input parameters names and types and the web service's returning data types.

C. Simple Object Access Protocol (SOAP)

The SOAP is considered as the de facto protocol for exchanging XML-based messages over the Internet. The descriptions of the web services and their request invocations and returned outcomes are represented as XML formats. SOAP uses the XML format for sending a request to and getting a response from the service provider application.

D. Extensible Markup Language (XML)

It is used for encoding web service descriptions, request invocations and returned values in a standard format readable form any operating system.

IV. Research Process & Pattern Identification

In a previous research we aimed to derive generic healthcare service patterns and their application on a real case study. We developed a prototype that resembled the generic services and we showed that the services are re-usable and customizable to suit several case conditions, as well as scalable to meet any level of consumer demand. [17] In this study we aim to expand the research and show how the proposed patterns can further improve orchestration and Web navigational access of public healthcare services over the governmental portal. The research process will utilize previous research results and will further develop them to derive orchestration and navigational models following the proposed patterns as shown in Figure 3.

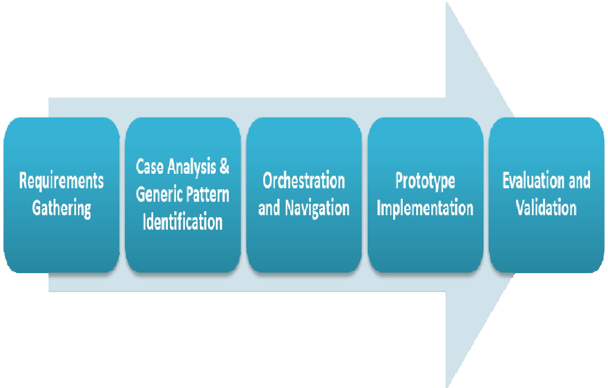


Figure 3. Research Process

A. Requirements Gathering

The aim of requirements gathering is to collect and analyze the requirements of the medical system, in addition to gather details on the patients to build the patients' medical history that will be utilized by data mining and clinical decision support activities.

Several interviews were conducted to collect the required data as these are the basis of the proposed prototype to reflect the needs of the doctors, patients, and the governmental hospitals. These interviews were conducted with the National Databases Program Director, and the project managers of the National Databases Program and technical specialists.

Different types of questions were covered to derive details on the public healthcare service process such as:

- What kinds of patient details the doctors will require to better prescribe medicines that suit patient history?
- What are the main clinical activities the public healthcare system (governmental portal) needs to support?
- In what ways does the doctor utilize the data to diagnose a patient case?
- What is important information in diagnosing a patient who suffers from more than one disease?
- How do governmental decision makers handle expansion of public healthcare services and building new hospitals across the different governorates in Egypt?
- Are there any decision mechanisms available to direct the distribution of the various medicines over the different governorates in Egypt?

The interviewees were acquainted with all the previous questions because they are working in the Ministry of State for Administrative Development which allows them to meet with different doctors and professionals. Also, the technical specialists have provided technical support to several hospitals and got hold of various hospital forms that contain the needed information of the patients.

B. Case Analysis & Generic Pattern Identification

It is about analyzing the collected data to specify the functionalities expected from the system. After analyzing the output of the interviews, it has been decided to model the requirements as services to cover public healthcare operations and beneficiaries/actors of the system. In order to decrease complexity of the system and specify generic abstractions of system operations, we have investigated the public healthcare value chain. According to Lawton Burns [2] the healthcare value chain consists of five main clusters: Players, Intermediaries, Providers, Purchasers, and Producers as shown in Figure 4. These clusters reflect the different actors in the system as shown in Figure 5; whereas each cluster has its specific role and typical operations/services.

Patterns are a cornerstone concept in modeling software architecture, as they aim to discover re-usable building blocks to improve flexibility and adaptability of software architecture, and at the same time attack complexity [3]. They help software architects to identify recurrence in functional elements and reduce redundancy in software design. Two common types of patterns are widely known; architectural patterns, that reflect structural attributes of software architecture, and design patterns that provide behavioral/functional representations of software architecture [8]. In this study the proposed patterns represent both types; firstly they provide conceptual/functional building block/services, secondly we propose that the implementation of these services to be broken down into layers to separate service interface from operations from data, etc. which provides architectural details.

C. Service Orchestration and Navigational Access

Orchestration in SOA reflects dynamic, flexible and adaptable adjustments of elementary services to meet the changing needs of the healthcare domain.[20] Orchestration encapsulates the basic services and integrates them through dynamic binding to form higher level processes of elementary composite services. Moreover, orchestration could be defined as a standards-based mechanism that defines how web services work together, to cover business logic, sequencing, exception handling and process decomposition by building on service and process reuse. [21]

Figure 4. Healthcare Value Chain

Public Health Care Value Chain				
Players	Intermediaries	Providers	Purchasers	Producers
•Government	•Insurance Companies •Pharmacies	•Hospitals •Clinics	•Patients	•Doctors •Nurses

Figure 5. Healthcare Actors and Generic Roles

The research will also address the navigational access design features that will take into consideration the principles of effective Web navigation and how the navigational access of public healthcare Web services can be improved through the proposed patterns. In the following we elaborate more on Web navigational access.

1) What does Web navigation mean?

Web navigation means going from one page to another that are provided by static hyperlinks or dynamic page search through the website. [5]

2) Why Web navigation is important?

Easy and effective Web navigation is important as it encourages users, especially those reluctant to use information technology, like in our case doctors, nurses, patients, etc. to use the website and reach the required functionalities in an easy to understand structure and layout. Ease of use and access of required services in least number of mouse clicks will help overcome resistance to utilize the new computerized system and avoid the tendency to return back to the manual system. [14]

3) How do we create an effective Web navigation?

First of all, we have to determine the website structure and organize it in a way to take into consideration the principles of effective Web navigation that will improve access to functionalities for the different users. According to usability studies that have been conducted, the Web navigation principles cover the following features:

a) Consistency

While designing the website consistency is important to prevent the user from being confused while using the website. Consistency of the website could be divided into: a consistent design (themes, images), consistent hyperlinks and organization that are related to the design of system architecture of concern in our study.

b) Help

It is important to take into consideration that users will need guidance while using the website. This can be supported by effective navigational design that allows reaching the diverse system functions in an easy way.

c) Pages navigation

The users will have better orientations through the website when they know which page they are using, how to go to the next page and how to return back to the previous page or menu.

d) Support and feedback

While designing the website, we need to notify the users if an error is encountered and how to overcome or deal with this error to navigate out of it.

e) Variety of navigation options

For a medical website with diverse users we should allow them to undertake their activities in different ways through multiple menus and a variety of access points, but still with consistency so that the user will not be confused.

D. Prototype Implementation

It is about designing the governmental healthcare service bus by building on the introduced patterns, and implementing a prototype to apply the patterns on the case study. As the proposed healthcare services are abstract and reusable the prototype will be generic for any governmental medical case. In implementing the patterns the prototype will build on the concept of "separation of concerns", where the Web service implementation will be divided into layers to ensure that the service details are loosely coupled which will induce flexibility on architectural level.

E. Evaluation & Validation

It covers the application of software metrics to measure the reusability, loose coupling, scalability and abstraction of the proposed healthcare service patterns. If the proposed patterns meet these quality attributes, then the patterns are validated. In addition we will apply different change scenarios to test the adaptability of proposed architecture following the patterns, as well as the scenarios of user navigations and how the patterns with provide an effective Web access.

V. Case Description & Application of Healthcare Service Patterns

The case study is one of the projects of the Ministry of State for Administrative Development in Egypt under the "System of integrated government services" that is concerned with system integration and transmission of information between different governorates. This should improve the communal services for the citizens. It aims to interlink the various governmental institutions and entities, and provide an integrated healthcare model under the guidance of the political leadership, which calls for improved coordination between public healthcare actors. This enables exchange of information in a secure way that will result in more accurate diagnosis and effective treatment with the use of integrated patient medical history data.

A. Specifying the Generic Public Healthcare Services

A service could be consumed by different users with different authorities, e.g. patient medical history service can be consumed by the doctor, nurses, patients and insurance companies.

The common, re-usable service patterns we identified through requirements analysis of the case study are shown in Table 1.

Players	Services Utilized by Players
Government	Disease cure history Patient citizen details Doctor syndicate details
Insurance Companies	Finance details of the patient Patient Medical history
Pharmacies	Medicine/Drug Details Disease cure history
Hospitals/Clinics	Disease cure history Finance details of the patient Patient citizen details
Doctors	Patient Medical history Disease details Disease cure history
Nurses	Patient Medical history Disease details Patient citizen details
Patient	Patient Medical history

Table 1. Generic Public Healthcare Service Patterns

B. Generic Service Patterns & Orchestration

There is an interaction between the different services to accomplish a single task for example adding a new patient to the system: First of all, the initial data of the patient should be retrieved from the civil registry through the patient/citizen service then the new patient's medical details should be added through patient/medical service as shown in Figure 6.

C. Separation of Concerns in Pattern Implementation

Implementing the different services and allowing them to interact with each other to build the complete medical system is done through the G2G service bus as each service is presented by a different governmental state as shown in Figure 7.

The benefit of separating systems into layers is to define the roles of each layer and to isolate the functions of each layer from other layers. This is because after some time the business backend, the business logic, the data format or the service contract/interface, or maybe all of them may change. If these layers are not separated, then changing in one of them will result in changing the whole service implementation. To avoid this chain of changes when designing web services it is recommended to separate these concerns by dividing them into layers.

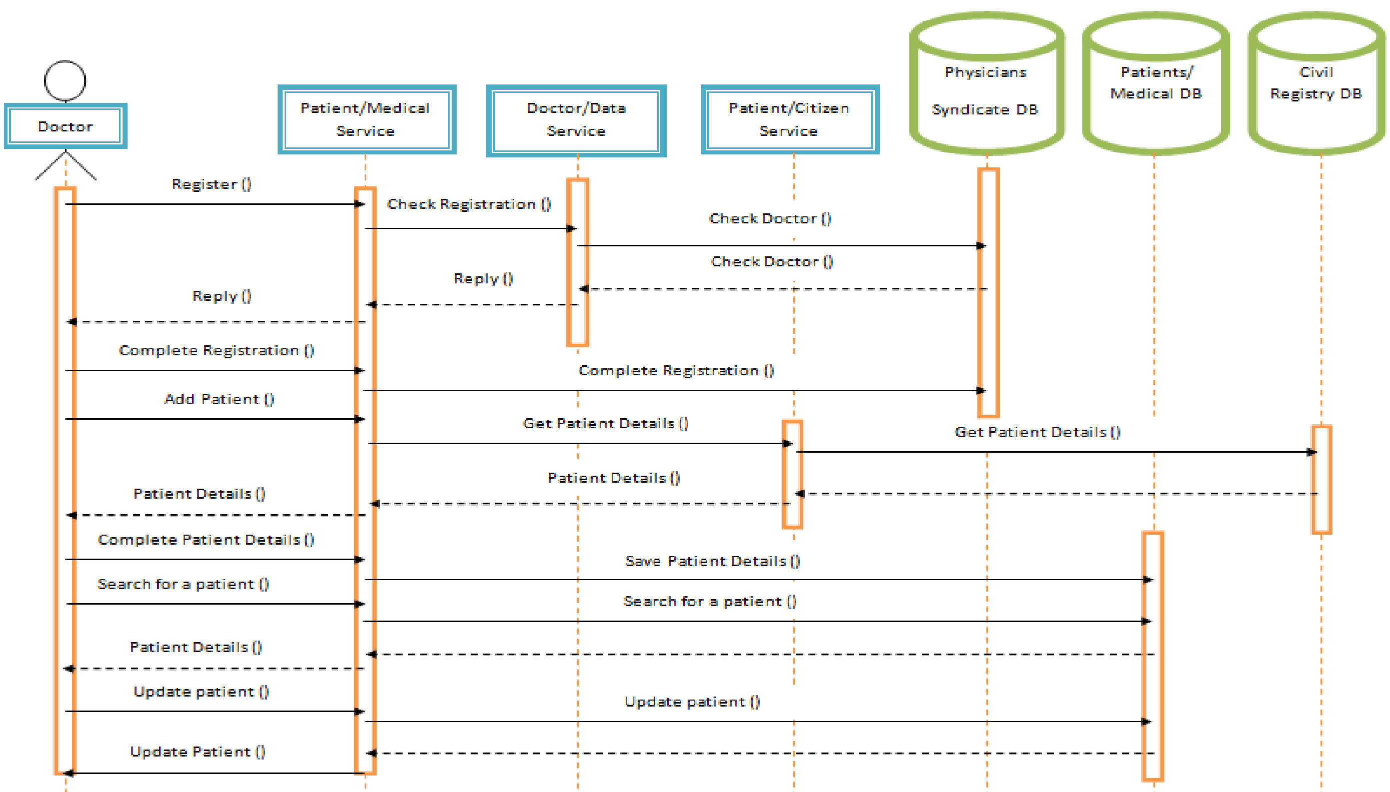


Figure 6. Orchestration between Patient Medical Data, Patient Citizen Data and Doctor Syndicate Data Services

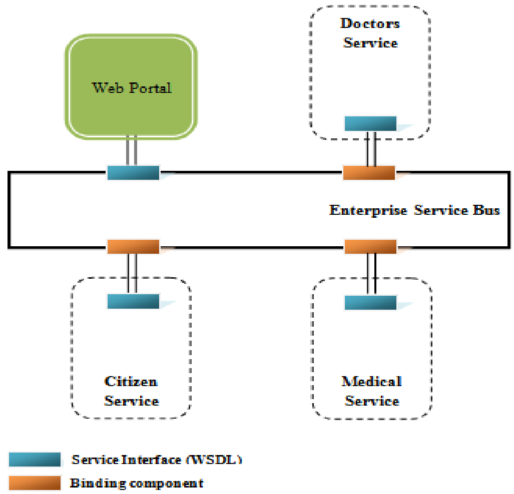


Figure 7. Public Healthcare Enterprise Service Bus

According to [15] the common known layers are; Business Layer, Data Layer, Logical Layer and the Service layer as shown in Figure 8.

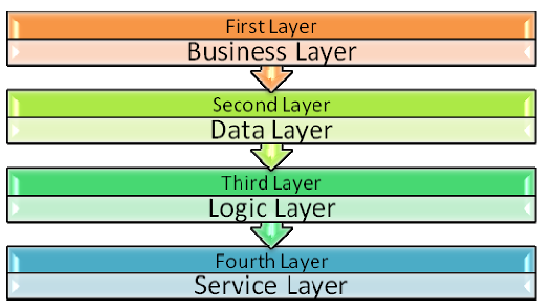


Figure 8. Service Layers

a) Business (Enterprise) Layer:

The business layer represents the enterprise system backend that covers components used to implement the enterprise activities and defines business entities that will be invoked by the service logic component, whereas enterprise data will be accessed by the data layer [15]. Thus, this layer is responsible of making sequence calls to data logic layer with authorization and validation, as well as implementing service logic required by the logic layer.

b) Data Layer:

It provides simplified access to data stored in any persistent storage media in the enterprise.

c) Logic Layer:

It handles the logical response to user interaction, e.g. update a field on the data layer, and also cover arithmetic calculations on data and other conditional logic to fulfill the service functionality. This will be reflected later on the enterprise service layer.

d) Service Interface Layer:

It contains a standard interface to invoke the required service by any consumer. Within the service layer, the service operation interface and the data contracts will be defined and implemented.

VI. Navigational Access Design Following the Patterns

One of the important aspects that should be taken into consideration while designing the Web portal is the level of users' knowledge of computer systems and how the different public healthcare players can easily use the automated system without guidance or training. To create an effective website, designers and users need to work together with a specific methodology to create a website that meets the requirements and encourages them to revisit the website. This requires that users feel comfortable, confident and satisfied using the application [5].

We have to put into consideration that the system should be easy to use and this can be achieved by applying the Human Computer Interaction (HCI) principles such as the ease of use, ease of access and layout principles. HCI principles that address color, visual aids, multi-media support, etc. or in other words presentation principles are out of scope of our research as we will focus on HCI issues related with architectural design and website navigation layout.

The patterns are responsible for defining reusable and loosely coupled system building blocks that will reflect on simplicity of architecture, and in turn simplicity of the navigational structure of the portal. This is because patterns are abstractions of Web portal functional usage and therefore reflect how the different users can utilize the various functionalities. Taking the patterns as guidance for navigational access will help remove redundancy in service representation and show how the different services can be utilized by the various actors that will reflect different points of access. Thus the patterns remove redundancy and represent different user aspects that will lead to simple and effective Web navigation. By adopting these principles the users will be less frustrated and make fewer mistakes [5]. HCI should seek "to understand and support human beings interacting with and through technology" [14].

The structure of website represents the content organization of underlying application. This organization is captured via the inter-page linkage between the pages, as reflected on hyperlinks. The organization also covers intra-page structure of the content within a page. All these elements represent the **functional usage structure or in other words patterns and their relationship with navigation structure of the application** [14]. The proposed patterns can improve the navigational access structure by defining common usage scenarios/functional elements and how users can reach/access these functional elements.

As shown in Figure 9 the different actors will utilize the same services for different purposes and in different ways. Thus the patterns will remove redundancy in navigational access and therefore will lead to simplicity and ease of use. For example, the doctor will access the portal through a user name and password that will be created for the doctor if his/her details are listed in the Physicians syndicate which will be checked through the Doctor Service. The nurse will access the patient details through the doctor account as she will be registered with the doctor as his/her assistant. The government will access the patient data via responsible employees that will have access only to limited information as the patient data are secured and sensitive. These data will be utilized to produce demographic statistics, e.g. death rate, life expectancy age, cure rate of certain disease, spread of a certain disease, etc. These data will help public healthcare decision making, for example defining new diseases and their spread across the governorates, choosing the right treatments for certain disease, distribution of health services over the governorates and their possible deficiencies, etc. Finally, the patient will have access to his/her personal data through the Civil Registry which will be done by invoking the Citizen Service.

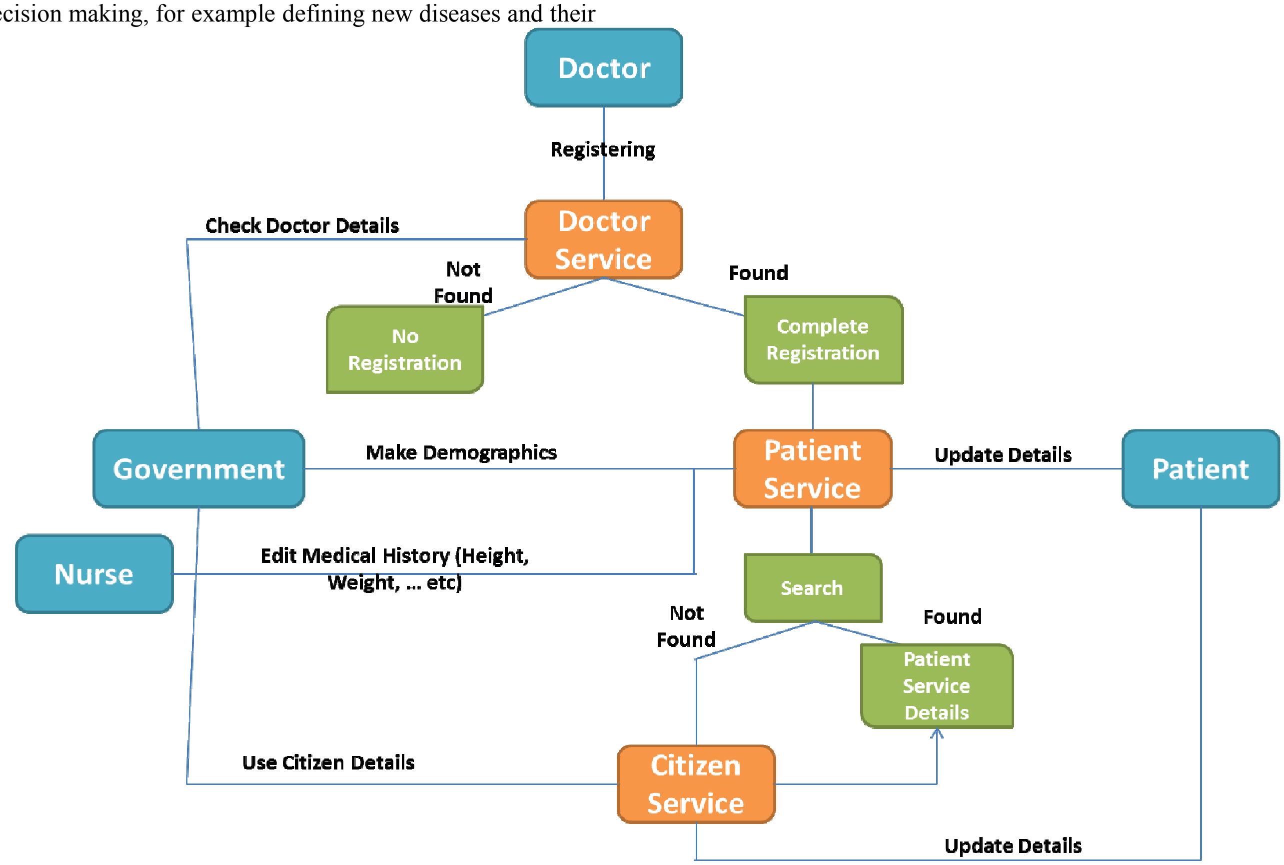


Figure 9. Navigational Structure

VII. RESEARCH VALIDATION

It is suggested to benchmark the impact of the use of the proposed patterns on the quality attributes of software architecture. Some common change scenarios are also considered to identify how the service layers will deal with change without affecting the other layers, and thus show how the change propagation from layer to another will be limited. Change scenarios are considered a profound technique to validate the flexibility and re-usability of software architecture [9].

A. The Impact of Proposed Patterns on System Quality

The major quality attributes and software metrics related to SOA are reusability, abstraction, loose coupling and scalability [13]. In the following we specify the impact of the use of generic service patterns on system quality, in particular how the software metrics of concern will be affected as shown in Table 2.

Software Metrics	Generic Service Pattern Impact
Reusability	The reusability of the system is proven as it covers generic services that can be consumed by different actors. The services can be also applied on different medical systems as they are generic.
Abstraction	Dividing the code into abstract functional elements/ services will allow the customization of the generic services to meet the differences in healthcare systems. This is possible by changing the content of the layer of concern to meet the specific requirements.
Coupling	Coupling within the service is decreased by applying the concept of separation of concerns as every layer is not related to the other layer; which means loose coupling between the different layers of service implementation. In addition coupling between the services is decreased as the services are abstract and re-usable, thus build on the concept of encapsulation and information hiding that decreases the coupling.

Scalability	Scaling up (adding new functionalities or serving more actors), as well as scaling down (removing some of the system functionalities or serving less actors) is possible because the services are generic and loosely coupled.

Table 2. Software Metrics for Pattern Validation

B. Change Scenarios & Implications on Service Patterns and Layers

Different change scenarios (Table 3) are considered and their impact on suggested service layers is conduced as follows:

1. Changes in the Business Rules:

It will lead to changes in the Logic layer with no changes in any of the other layers.

2. Reuse of Service:

The service could be utilized by different actors; for example the Medical Service can be reused by the doctor, the hospital, the insurance company and many other actors.

3. Scaling Up and Scaling Down:

Adding more functionalities or removing some of the system functionalities will lead to changes in the Service layer only, provided that the services have been already implemented. Else the logic, data and business layers will be implemented to meet the new needs. As long as the system is used for many years, more services will be implemented incrementally.

4. Changing the Backend/Enterprise:

Changing in the syndicate system platform will not affect the service, provided that it still implements the required functionality. The data layer might change if the syndicate data will change, but this is rare, as the changes are always in upgrading the platform and adding more functionality without altering the data.

Change Scenarios	***Impact on Service Layers***			
	Business	***Data***	***Logic***	***Service Interface***
Changes in Business Rules	No	No	Yes	No
Reuse of Patient Medical History Service	No	No	No	No
Scaling up and Scaling down	No	No	No	Yes
Changes in the Backend	Yes	Yes/No	No	No

Table 3. Change Scenarios & Impact on Service Layers

C. Scenarios of Navigational Access

In the following we investigate different scenarios of navigational access for invoking the same service and how the patterns unify the access although service consumers and their rationale vary.

1) Amending or Updating Patient Details:

The patient details could be accessed by different users that will invoke several changes on the same service as follows.

a) Doctor

After diagnosing the patient, the doctor will add the new diagnosis, prescribed medicines … etc.

b) Nurse

The nurse may change the patient details such as the weight, medical status, etc.

c) Government

The government may need to change some basic information about the patient such as the address … etc.

2) Using Medical History

The patient medical history could be accessed in different ways:

a) Doctor

To diagnose the patient case, the doctor needs to know the complete medical history of the patient in order to give him/her the right treatment with the right dosage.

b) Pharmacies

The pharmacy will check the disease cure rates and the used medicines to examine the medicine cure effectiveness.

c) Government

The government may need to identify demographic aspects, such as outbreak of special diseases, their spread across the governorates, cure rate, etc. This will require invoking the medical history service of patients that suffer from this disease.

Thus the proposed patterns reduce redundancy in navigational access in spite of diversity of actors and their rationale in calling the same service, which will make the navigational access simpler and more organized. This will lead to easy to follow navigation, better understandability of Website structure and speed in reaching the required information that are important features of navigational design as discussed earlier. In addition patterns induce consistency of service representation that also improves navigation.

VIII. Windows Communication Foundation (WCF) & The Practical Application of the Proposed patterns

According to Microsoft, Windows Communication Foundation (WCF) is a framework for building service-oriented applications. It enables developers to build secure, reliable, transaction-based solutions that enable integration across various platforms and thus induce interoperability that will make utilization of existing investments possible.

A. WCF Components (ABCs)

There are many terms and concepts surrounding WCF such as address, binding, contract, endpoint, behavior, hosting, and channels [11]. In the following we will elaborate more on these concepts.

1) The (A) stands for Address

The WCF address indicates the destination of the service provider where a message request will be sent. It is a URL, with the first part specifying the transport mechanism, and the second hierarchical parts specifying the unique location of the service.

2) The (B) stands for Binding

Binding is important for the communication of the service call between the clients and service providers by building on computer network techniques, such as transport, encoding, and verification. WCF uses binding to generate the underlying wire representation of the endpoint.

3) The (C) stands for Contracts

The interfaces of WCF Service are defined by so-called contracts. A WCF service communicates with other applications according to its contracts. There are several types of WCF contracts such as Service Contract, Operation Contract, Data Contract, Message Contract, and Fault Contract, each specifies a certain perspective of service description.

a) Service Contracts (Maps to WSDL)

A service contract is the title of the Service because it tells what could be done with the service. It mainly includes service-level information, such as the name of the service, the namespace of the service, and the corresponding call-back contracts of the service. Methods or service operations that are used for specific tasks could be defined inside this interface. Normally, a WCF service has at least one service contract.

b) Operation Contract

An operation contract is defined within a service contract. The parameters and return variables of an operation are defined in the Operation Contract. Different data types could be used such as primitive data type (i.e. integer), or composite data types (i.e. date, bank account). For a service to work as a WCF service, the operation contract needs to be implemented. It is also responsible of defining operation-level settings (i.e. the transaction flow of the operation), the directions of the operation (i.e. one-way, two-way, or both ways), and the fault contract of the operation to handle exceptions.

c) Data Contract (Maps to XSD)

The data types of the WCF services are defined in Data Contracts. Metadata are important to describe the data types used by the WCF service with the aim to enable other applications to interoperate with the service. A data contract can be used by an operation contract as a parameter or return type, or it can be used by a message contract to define elements of messaging. If the data types of the operations are primitive, there is no need to define any data contract.

d) Message Contract (Maps to SOAP)

Message contracts represent request or response information that an operation contract will pass as a parameter or a return message. A message contract is responsible of defining the parameters of the message, and the message's security level.

e) Fault Contract

The caller of a service operation contract should be warned of any error or faults that could return to it. These error types are defined as fault contracts. The number of fault contracts that can be associated with an operation can vary between zero and more.

B. WCF Endpoints

Messages are sent between endpoints to define the information required for the message exchange. A service can expose this information as the metadata that clients process to generate the appropriate WCF clients and communication stacks. A WCF service endpoint has an address, a binding, and a service contract.

1) Endpoint Address

It is a network address to describe where messages should be sent. Normally, one unique address is assigned to each endpoint, but sometimes it could be shared by two or more endpoints.

2) Endpoint Binding

It specifies the transport protocol (TCP, HTTP), encoding (text, binary), and security requirements (SSL, SOAP message security) to describe how the endpoint communicates with the world.

3) Message Exchange Contract

It is a collection of organized messages to specify what communication types the endpoint delivers. Its basic Message Exchange Patterns (MEPs) could be one-way, duplex, or request/reply.

It can be concluded that the Windows Communication Foundation made the idea of patterns applicable as it breaks down the implementation of the service into several aspects which will make services more generic and more flexible to deal with change. Also, WCF embraces the idea of separation of concern that could be further improved by applying service layers as suggested in this study, which will further improve scalability and flexibility of service implementation.

IX. CONCLUSIONS

The analysis of the Egyptian public healthcare case study indicates that several parts of the governmental systems are still working manually or not fully automated with limited integrated functionalities. For example, many public hospitals do not still have computerized database of their patients where these data are stored on papers. Invoking the healthcare services in a manual way will lead to waste of time in preparing, processing and transmitting information. This leads to delays in healthcare services and inaccurate diagnosis and results.

Service Oriented Architecture (SOA) is the next generation programming paradigm because objects are encapsulated as autonomous services, each with its own contracts (interface definition) and unique functionality. With the use of SOA we can help hospitals implement their services by building on their legacy systems whatever the software platform they use as service contracts are standardized and thus will make inter-operability possible. Web services are loosely-coupled and therefore manual activities can be incrementally automated and added to the enterprise service bus ESB. As SOA provides a standardized data bridge, integration and interactions between hospitals become easy to realize, e.g. to collect complete patient medical histories from several hospitals that would help doctors better diagnose the patient case and prescribe accurately the right medicines.

Patterns are an important concept in design of software architecture with the aim to attack complexity, leverage development productivity and improve system performance through consistency and redundancy removal. The proposed patterns are generic and loosely coupled, hence can be customized with minimum efforts to suit different health care systems around the globe. They are also scalable where it is possible to implement some healthcare activities without the need to implement the whole healthcare value chain. In particular patterns are known to induce flexibility through the re-use of generic building blocks and thus will improve maintainability and extensibility of software architecture to deal with future needs. It is suggested that the proposed service patterns and service layers will support the design of generic, loosely coupled public health care systems that will simplify their representation and thus improve their navigational access.

X. Future Work

Medical systems are complex as they cover diverse and distributed parts that serve different users and actors. Our research aimed to prove the applicability of the proposed patterns to attack complexity and improve flexibility and inter-operability of healthcare systems. Our scope covered the doctor and the patient patterns. Many other patterns need to be added to the system on the long run to serve all different players. A flexible, efficient health care is an important topic that requires special care.

As the patterns induce flexibility and extensibility further add-ons can be implemented in future, such as data mining and business intelligence services that would support diagnosis activities, as well as decision support to governments in outlining public healthcare policies without the need to re-build the architecture. In future, the patterns will be also applied on the level of clinics to collect patient medical history from all possible resources which will improve the accuracy of the diagnosis process.

Future work will also focus on the use of SOA within a cloud computing architecture where the healthcare portal resources will be virtualized to allow better utilization of resources and optimized data processing time. The patterns will be implemented as Software as a Service (SaaS), as well as the data mining services to help the decision support for the government, doctors and pharmacies

References

[1] Bloomberg J., ZapThink, 2004, "The SOA Implementation Framework: The Future of Service Oriented Architecture Software" (April, 2004). Retrieved from: http://www.inst-\informatica.pt/servicos/informacao-e-documentacao/biblioteca-digital/arquitectura-e-desenvolvimento-de-aplicacoes/soa/soa.htm.

[2] Burns L., DeGraaff R., Danzon P., Kimberly J., Kissick W., Pauly M., *The Wharton School Study of the Health Care Value Chain*. Published by Wharton School, United States.

[3] Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M., 1996, "Pattern-Oriented Software Architecture: A System of Patterns". John Wiley and Sons, Chichester

[4] Cavanaugh E., Altova, 2006, "Web Services: Benefits, Challenges and a Unique Visual Development Solution"

[5] Issa T., Turk A., 2012, "Applying Usability and HCI Principles in Developing Marketing websites", *International Journal of Computer Information Systems and Industrial Management Apllications*, Vol. 4 No. 9, pp. 76-82

[6] Kart F., Miao G., Moser L., Melliar-Smith P., "A Distributed e-Healthcare System Based on the Service Oriented Architecture" (2007). *IEEE International Conference on Services Computing (SCC* 2007)

[7] Kim H., Lee R., MS2Web: "Applying MDA and SOA to Web Services, Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing Studies in Computational Intelligence", 2008, Volume 149/2008, pp. 163-180.

[8] Koch, N., 2009, "Modelling Service-Oriented Systems with UML4SOA". In: 3rd SENSORIA Summer School (SENSUS 2009), Keszthely, Hungary

[9] Kruchten P., 1995, "Architectural blueprints – the 4+1 viewmodel of software architecture", *IEEE Software*, Vol. 12 No. 6, pp. 42-50

[10] Kumar A., Ranjan M., "Service Based System Monitoring Framework", *International Journal of Computer Information Systems and Industrial Management Applications*, Vol. 3 No. 104, pp. 924-931

[11] Liu M., *WCF 4.0 Multi – Tier Services Development with LINQ to Entities*. Birmingham: Packt Publishing Ltd.

[12] Mahmoud Q.,2 005, "Service Oriented Architecture (SOA) and Web Services: The Road to Enterprise Application Integration (EAI)" Retrieved from: http://www.oracle.com/technetwork/articles/javase/soa-142870.html

[13] Mansour Y., Mostafa S., 2011, "Assessing Internal Software Quality Attributes of the Object-Oriented and Service-Oriented Software Development Paradigms: A Comparative Study" (April, 2011), *Journal of Software Engineering and Applications*, 2011, 4, 244-252.

[14] Martinez G., Rodriguez F., 2011, "Using User Interaction to Model User Comprehension on the Web Navigation", *International Journal of Computer Information Systems and Industrial Management Applications*, Vol. 3 No. 99, pp. 878-885

[15] Microsoft, *Business Layer,* Retrieved from: http://msdn.microsoft.com/en-us/library/ff699423.aspx

[16] National Public Health Partnership, 1998, "Public Health in Australia: The Public Health Landscape: person society environment", *Australia Report* (Melbourne, NPHP).

[17] Nazih M., Alaa G., "Generic Service Patterns For Web Enabled Public Health Care", *IEEE Xplore, Next Generation Web Services Practices (NWeSP) 2011*, pp. 274-279

[18] SOA – Terminology Explained, 2010. Retrieved from: http://www.lemongrassconsulting.com/knowledge/terminology-explained .

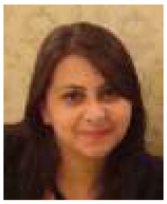
[19] Tchuta L., Chunhua G., 2011, "Atomic New Service Pattern", *International Journal of Software Engineering and its Applications*, Vol. 5, No. 1, January 2011

[20] ToolboxIT, "Why Orchestration defines your SOA", 31 March 2005. Retrived from: http://it.toolbox.com/blogs/davidlinthicum/why-orchestration-defines-your-soa-3685

[21] ToolboxIT, "SOA = Orchestration", 8 September 2005. Retrieved from: http://it.toolbox.com/blogs/davidlinthicum/soaorchestration-5670

Author Biographies

Marina Nazih was born in Cairo, Egypt. She studied at Collège du Bon Pasteur and is a graduate of the British University in Egypt (BUE) in 2011, Faculty of Informatics and Computer Science with Specialty in Software Engineering where she scored the highest rank, excellent with Honors. This research covers her graduation project that was carried out in partnership with the Ministry of State for Administrative Development, Egypt.



Ghada Alaa was born in Cairo, Egypt. She is currently an Information System Specialist at the Egyptian Cabinet's Information & Decision Support Center. Prior to that she was Lecturer at the British University in Egypt for four years. She holds a Ph.D. in e-Commerce Systems Development, Brunel University, UK (2006) and a B.Sc. in Biomedical & Systems Engineering, Cairo University, Egypt (1995). Her research interests lie in the area of Web2.0 and Service-Oriented Architecture design patterns.