Developing A Service Oriented E-Government Architecture
Towards Achieving E-Government Interoperability

Zuhoor Al-Khanjari, Nasser Al-Hosni and Naoufel Kraiem

Department of Computer Science, Sultan Qaboos University
PO Box 36, Al-Khoud 123, Oman
{zuhoor, m023581, naoufel}@squ.edu.om

Abstract

E-Government offers a new electronic channel by means of which citizens and government ministries can interact with one another, unconstrained by the locations and schedules; and thereby improving government effectiveness. However, realizing this vision is strictly depending on the ability of diverse computing systems owned and managed by different government ministries to be able to interact together across all ministerial boundaries. This ability is known as e-government interoperability.

Since ministries have built their computing systems independently with specifications and solutions relevant to their particular needs but without adequate attention to the need to interact with other ministries systems, this has result in a patchwork of heterogeneous computing solutions that have limited coherence and are largely uncoordinated. Therefore, during the last few years, e-government interoperability has been an important research area. To this direction, although several approaches have been proposed, all these approaches would be insufficient since they are theoretical solutions and technically focused.

In this paper low level e-Government architecture is proposed driven by Service Oriented Architecture (SOA) to solve the architectural problem in achieving seamless e-government interoperability. This architecture acts as unifying architectural vision and helps in finding a common understanding of e-government and its realization.

Keywords: E-Government, interoperability, Service Oriented Architecture, Service Oriented e-Government Architecture

1. Introduction

Information technology has extremely changed the way of doing business and performing related transactions[1]. It has noticeably dominated ordinary citizen’s everyday life [2]. Towards this direction, recently many governments across the world started their own initiatives to use computing systems to offer a new electronic channel by means of which citizens, and government ministries can interact with one another through automating public services which can allow citizens to more easily access all kinds of government services in an automated way that are unconstrained by the locations and schedules; thereby improving government efficiency and effectiveness.

This initiative to use Information and Communication Technology systems (ICTs) to automate government services is called “e-Government”. This term e-Gov emerged in the late 1990 [3]. Like any other modern concepts, e-Government has multiple definitions among researchers and specialists.

However for the purpose of this study, we define e-Government as: The delivery of government services in automated manner using computing systems. From this definition and many other definitions in the literature we can draw our first conceptual rule of e-Government which states that: “e-Government exists if and only if computing systems exist”. Otherwise, e-Government is more than existence of a computing system, website, email or processing transactions via the internet. E-Government is an exact virtual world
of the government physical world which actually has a dynamic mixture of goals, structures and functions [4] that are working together seamlessly in agreed upon fashion. Therefore, the success realization of e-Government is not only based on the existence of computing systems but it is strictly based on the ability of these diverse computing systems owned and managed by different government ministries to be able to interact together across all ministerial boundaries seamlessly. This ability known in the literature as an e-Government interoperability.

E-Government interoperability, in its broad sense, is defined as the ability of ministries to work together. At a technical level, it is the ability of two or more diverse government computing systems or components to meaningfully and seamlessly exchange information and use the information that has been exchanged [5].

We define e-Government interoperability as the ability of different (distinct, heterogeneous, silos) computing systems owned and managed by different ministers (organizations, authorizes, agencies) to interact semantically & effectively to conduct transactions in an automated manner. The word interact used is the definition has three possible meaning. First it mean the ability of these systems to exchange information. Second, it means the ability of one system actions to be triggered by another computing system. Third, the ability of one system to invoke a service on another computing system that generates either a synchronous or an asynchronous response.

Interoperability reduces or eliminates the problems of islands of automation. It enables business processes to flow from one application to another. Interoperability enables one system to work with another, in near real-time fashion, to share critical business information. So from these facts all together, we can draw our second conceptual rule of e-government which states that: “e-Government successes if and only if existing computing systems are interoperable”.

However, e-Government interoperability is not an easy task to achieve. It has been recognized as a key challenge and crucial issue for e-Government at least since 2001 [6]. This is because, realizing e-Government interoperability is hindered by difficulties connected to implementation. These difficulties or challenges are faced because initially government ministries have built their computing systems independently with specifications and solutions relevant to their particular needs but without adequate attention to the need to connect, exchange and re-use data with other ministries systems. This results in a patchwork of heterogeneous computing solutions, that have limited coherence and are largely uncoordinated [5].

Moreover, as more and more complex information systems are put into operation every day, the lack of interoperability appears as the most long lasting and challenging problem for governmental ministries which emerged from proprietary development of applications, unavailability of standards, or heterogeneous hardware and software platforms [7] and [8]. Therefore, during the last few years, e-government interoperability has been an important research area in order to facilitate the seamless exchange of information across governmental ministries’ computing systems.

Nevertheless, there are multiple, interrelated interoperability issues which affect software implementation of e-Government projects. These issues must be identified and addressed before an IT implementation for e-Government can be developed.

To this direction, there are various approaches which have been proposed and adapted by many governments towards approaching their e-Government interoperability such as adaption of standards and architectures. However, merely following those approaches would be inadequate since all these approaches are considered only as theoretical based guidelines that hindered with many barriers and obstacles to adapt. Therefore, there is a need for further research to find more practical solution to solve this architectural problem to successfully implement e-Government interoperability.

A Service Oriented e-Government Architecture is presented in this paper to solve this problem based on the fact that e-Government is a single coherent computing system made
up of distinct computing systems interacting together that lacks whole e-Government low level system architecture. This proposed architecture represents a holistic view of an interoperable e-Government vision.

In the paper the relevant literature reviews of e-Government interoperability approaches are presented first. Then, the proposed architectural solution (Service oriented e-Government Architecture) is presented and described in details with its promised benefits in the Section 2. Finally, the future work and conclusions are presented and discussed in Section 4.

2. Literature Review

First steps towards achieving e-Government interoperability were taken at early 1980’s when standardization in ICT started as a typical response to concerns related to proprietary systems [9]. This approach of adapting standards known as Electronic Government Interoperability Framework or e-GIF [10]. The e-GIFs are defined as a set of standards and guidelines that set out a common language to ensure coherent flow of information across systems [5]. In many countries, governments have developed their own e-GIFs like UK e-GIF [11], European Interoperability Framework (EIF) [12] and Australian Government Technical Interoperability Framework (AGTIF) [13] (see Figure 1).

![Figure 1. eGIFs in European Union [14]](image)

However, adaption of standards does not guarantee interoperability. For example, to date, the UK eGIF which is one of the most mature national interoperability frameworks: its first version was published in 2001, and it had reached version 6.1 as of March 2005, yet the interoperability problem remains [14].

The other approaches to achieve e-Government interoperability were through the adaption of architecture. The relevant architecture to Electronic Government Interoperability is known as Enterprise Architecture (EA), specifically National Enterprise Architecture (NEA). The EA stresses the planning and management of all IS assets and their architecture together with organizational structures and processes [15]. According to Christiansen and Gotze [16], 67 percent of governments have established an EA program and many are planning to do so in the near future [16]. The main pursued goal of adapting EA is to improve efficiency, and cross-government interoperability [17]. Furthermore EA itself does not provide any solution models to achieve interoperability but it rather addresses the strategic and functional requirements of all ministries (independently and collectively) and potentially points out the areas in which these can be implemented[17].

Recently, this approach influenced by Service Oriented Architecture (SOA), results in NEA/SOA approach which has the same goal as NEA. As the scope of SOA has expanded, a modern SOA is more about thinking in terms of service orientation and less
about implementation. The scope of SOA goes well beyond software architecture[18] and [19]. Ibrahim and Long in 2007 mention similarities between EA and SOA domains are that they both address similar architectural domains, are intended to closely align IT with business, use input based on business objectives, and require similar strategies and planning activities [20]. The challenges faced by governments implementing an NEA/SOA are the same as those countries implementing the eGIF [5].

There are other approaches proposed as well to achieve e-Government interoperability such as: second generation of eGIF which basically the same as eGIF, but it is provided with supporting tools to help in implementing eGIF standards [21] and [7]. Also, hybrid eGIF-NEA approach where those who adopt this approach makes no difference between eGIF and NEA. Germany’s Standards and Architecture of e-Government Applications (SAGA) version 2 is an example of a document that contains both the architecture and standards for e-Government interoperability [22].

To summarize the literature, it is easy to publish e-GIF, NEA, 2nd generation of eGIF, hybrid eGIF-NEA and NEA/SOA strategies, and mandate government ministries to comply with them, but that would not solve the problem, because of many reasons:

- No compliance enforcement: There is no guarantee that ministries will truly follow [23].
- Inadequate understanding: The strategies details are poorly understood by the ministries stakeholders.
- Inadequate capabilities: Ministry development teams are missing the technical capabilities to implement the technical standards. (For example, SOAP/ web services… etc.).
- Bureaucracy: “Many governmental departments have entrenched cultures which avoid openness and cooperation with others”[21] and [7].
- Not practical: All these approaches are theoretical/technical based documents.

Therefore, merely following those approaches would be inadequate[23], [21] and [9]. Figure 2 presents these approaches. Since all these approaches are considered only as guidelines that can help to achieve e-Government interoperability, the successful implementation for e-Government interoperability needs more practical and systematic approach.

**Figure 2. Summary of e-Government Interoperability Approaches**

In this paper a more practical solution is proposed driven by software engineering based on the judgment that e-government is in fact a single coherent computing system consisting of distinct computing systems interacting together. However, this vision lacks that whole e-Government low level system architecture which is proposed in this paper to solve this architectural problem in achieving seamless e-government interoperability. This proposed architecture is driven by SOA and called “Service Oriented e-Government Architecture”.

---

**Considered as guidelines & concepts that are theoretical and/or technical based.**

Architectures (NEA) Standards (eGIF)

(NEA/SOA) 2nd Generations

Hybrid NEA/eGIF
3. Service Oriented e-Government Architecture

In this section the proposed architectural solution to help achieving interoperability between heterogeneous independent ministries’ computing systems is presented and described.

3.1. As-Is System

Before identifying the proposed architectural solution which is considered as “to-be system”, it is important to define the “as-is system” i.e. the current status of the existing computing systems. The existing status result from an independent initiatives of government ministries to build their own computer systems with specifications and solutions relevant to their particular needs. Figure 3 below depicts these initiatives scenario.

These investments of utilizing computer systems in government sector result in: isolated, independent, heterogeneous computer systems that have limited coherence and largely are uncoordinated.

3.2. To-Be System

A key determinant of success in e-government initiative is based on the ability of these isolated, independent, heterogeneous computing systems to cooperate and work together sharing information and integrating processes across all boundaries surrounding their isolated computing systems. This requires a holistic view (whole e-Government low level system architecture) of the “to-be system” that would combine and organize these existing systems under one umbrella.

In this section, this proposed “to-be system” architecture is presented and described. This proposed architecture shall take into considering existing independent software
systems deployments and technical implementation rather than trying to replace them. Ministries have already invested heavily in building their running systems. It would be impractical to suggest a big bang approach that induces fundamental changes on existing infrastructures.

The proposed architecture is shown in Figure 4 below. It is consist of three main components. The first main component is (numbered as No.1 in the Figure 4) called Service Oriented e-Government Architecture. Second component is (numbered as No.2 in the Figure 4) called Interoperability Repositories. Third component is (numbered as No.3 in the Figure 4) called Development and Quality Assurance Infrastructure. The whole architecture named with the first component name as it is the most important component.

3.2.1. Service Oriented e-Government Architecture: This main component is driven by a service-oriented architecture (SOA). This is because SOA represents functionality of software applications as services with the goal of establishing native interoperability within services. Interoperability is specifically fostered through the consistent application of design principles and design standards. This establishes an environment wherein services produced by different projects at different times are intrinsically interoperable and can be repeatedly assembled together into a variety of composition configurations to help automate a range of business tasks regardless of when and for which purpose they are delivered[24].

Other characteristics to SOA services are that they are well defined, self-contained and abstracted to be coarse-grained [17]. Moreover, these services are designed to be reusable and loosely coupled building blocks to enable building of new business processes, that
can be created by combining the existing services located in a service repository. As web-services technologies enable standardized platform-independent interactions over the internet, ministries can easily employ the services other ministries offer and likewise offer their own services to other ministries. All these characteristics makes SOA the best architecture match for e-government integration ([2, 17]and [22]). Sanati and colleagues in 2007 stated that stated that: “We believe that government agencies need to implement SOA, as it is the best possible architectural design pattern suitable for integrating their e-services”. Where Emmanuel stated that: “It has been suggested that SOA is the best underlying paradigm with which to develop e-government services that can be used in cross-agency and cross-border situations” [25].

Figure 5 depicts only these layers of the proposed Service Oriented e-Government Architecture. This main component is divided into five layers (ordered bottom up): Operation layer, Semantic layer, Services layer, Process layer, Presentation layer.

![Figure 5. Main Component of the Proposed Architecture](image)

a) **Presentation layer**: Through this layer the end users (citizens or residents, or government employee) can interact with government public services. This layer can be rich-client, mobile, portal, or Web-based to invoke business processes (public services).

b) **Process layer**: In this layer the business processes are created and realized by the composition of business services which are implemented as a service (a piece of code performing well defined task). As defined by Davenport [26], business process is “a set of logically related tasks performed to achieve a defined business outcome”[26]. These related tasks constituting the business process are bind together in agreed-defined manner and executed in this layer.

c) **Service layer**: The tasks’ functionality residing in this layer are provided as services. Where these services are provided on purpose by the legacy systems within the ministries to acquire the requested information or perform a business task. The services within this
layer may be re-used and recomposed by different business process to automate different transactions. Services are at the heart of this multilayer architecture and are components that realize service flows and processes.

d) **Operation layer**: The existing legacy computing systems owned and managed by ministries constitutes the operation layer.

e) **Semantic layer**: Semantic layer falls between the operation layer and service layer to ensure that all the exchanged information between different legacy systems by services are represented and understood in within the same context and meaning. This layer may contain: metadata, ontologies, standardized data such as shared look up tables... etc.

### 3.2.2. Interoperability Repositories

The second main component of the proposed architecture is what we call Interoperability Repositories. It is composed of three repositories (business process repository, Services repository, Semantic Centre) as shown in Figure 6.

a) **Business process repository**: It contains details of business processes (public services) that are executed in the process layer. These business processes can be modeled using UML diagrams.

b) **Services repository**: It works as services registry (a database) storing the descriptions of all available government services provided by all government ministries. These services are registered in this repository after they are approved by the e-Government center of excellence. (see later).

c) **Semantic center**: It is considered as means to provide meaning-centered of all data exchanged between ministries legacy system.

### 3.2.3. Development and Quality Assurance Infrastructure

Third component of the proposed architecture is considered as a development infrastructure that is required to help in developing and automating new public services and it also ensures the quality of these new public services and ensures the reusability of the existing services. This component consists of the following as despite in Figure 7.
Figure 7. Development and Quality Assurance infrastructure

a) E-Government Centre of Excellence: This committee of the e-Government plays an anchor role in managing the interoperability repositories. This committee ensures the quality of the services, process models and semantic references added to the repositories. It enforces the adaption of standards published in the eGIF. Also addressing the approval and activating new services and retiring services that are no longer needed. This committee can be classified into groups such as:

- Academic: This group provides guidelines on how e-Government projects should be approached but doesn't itself conduct any detailed activities on modeling, development, or deployment.

  They look at design and other aspects from a quality assurance perspective and might have enforcement powers in this regard. They might supervise the training and certification programs.[27]. They could also conduct research into upcoming technological trends with regard to the e-Government growth.

- Advanced Technical: Knowledgeable and experienced business and technical members from any ministry participate in the Center of Excellence activities. They have wider powers of enforcement than academic with regard to design decisions, best practices, quality control, and testing approaches. This group is also responsible to take approval decision regarding interoperability repositories content.

b) E-Government Development Team: This is a central development team of e-Government public services. It consists of three main groups: business, legal and technical specialists. Those groups’ members are considered as a shared HR specialist across all government ministries. They are knowledgeable and experienced to work hand by hand with other similar groups in ministries to guide them to practice the best practices. Also to provide them with advanced and customized support to help them in developing the required services to automate the public services.
c) **Knowledge System:** this system is designed as a part of this architecture in order to support e-Government development team for getting interoperability knowledge as well as experiences. It contains the best practice cases such as a successful implementation cases in e-Government interoperability. It also contains the failed cases in order to help the development team to avoid the reasons behind.

d) **Discovery System:** this is one of the core engines of this architecture, its main role is to attend development team enquiries, finding and recommending the best available service for a given task. This engine should provide simple search APIs or web-based GUI to help find services.

### 3.3. The Advantage and Benefitsof the Solution

As a result, this proposed architecture will provide many advantages and benefits towards approaching e-Government interoperability. These advantages and benefits includes:

1- This architecture is considered as a depiction of the e-Government –low level system architecture - aids in the understanding of how the big system will look like and behave.

2- This architecture serves as the blueprint for both the e-Government big system and the project developing it, defining the work assignments that must be carried out by design and implementation teams.

3- It significantly indicates the e-Government system qualities such as performance, modifiability, and security. None of which can be achieved without a unifying architectural vision.

4- This architecture is also an artifact for early analysis to make sure that a design approach will yield an acceptable system. By building effective architecture, you can identify design risks and mitigate them early in the development process.

5- Last but not least, it helps architects from all over the ministries management to find a common understanding of e-government and its realization.

### 4. Conclusions and Future Work

In this paper the existing -so far- approaches in the literature to tackle e-Government interoperability challenge is presented and described – Figure8: Literature Review –. Nevertheless, this paper tackled this challenge from software engineering perspective by realizing the fact that e-Government is a big computing system that consists of its lower level view of a collection of distinct, heterogeneous computing systems – Figure8: As-Is System – interacting together seamlessly even they are managed and owned by different ministers. This view lacks that whole e-government low level system architecture which is one of the main architectural problem in achieving seamless e-Government interoperability.

This paper proposed an architectural approach to implementing electronic government interoperability. The approach is called “Service Oriented e-Government Architecture” – Figure8: To-Be System – and consists of three main components: service oriented layer architecture, interoperability repositories, and a development infrastructure. This proposed architecture works as holistic view of an interoperable e-Government vision.
As a result, this proposed architecture would provide many advantages and benefits that may add value towards achieving e-Government interoperability. This architecture serves as a depiction of the e-Government – low level system architecture - aids in the understanding of how the big system will look like and behave. Moreover, this architecture serves as the blueprint for both the e-Government big system and the project developing it, defining the work assignments that must be carried out by design and implementation teams. Last but not least, it helps architects from all over the ministries management to find a common understanding of e-government and its realization.

Future steps of our work include the development of a generic development methodology to develop and automate e-Government public services in systematic way – Figure 8: Future work –. This generic methodology will be restricted with some constraints for the purpose of realizing the proposed architecture in the paper. Then, a case study will be developed to evaluate the applicability of our proposed architecture & methodology.

References


Authors

Dr. Zuhoor Abdullah Salim Al-Khanjari has over 20 years of experience in teaching and research. She is currently HoD, Department of Computer Science (DCS), College of Science, Sultan Qaboos University (SQU), Muscat, Oman. She is Associate professor in Software Engineering (SE). She received her B.Sc. in 1993 in Maths&Computing from SQU, and both M.Sc. in 1995, and PhD in 1999 in Software Engineering from the University of Liverpool, UK. She is Chairing the SE Research Group,. She Chaired the E-Learning in DCS and College E-Learning Committee. She was the College Assistant Dean for Postgraduate Studies and Research during 2003-2006. She participates in the Evaluation Committees for IT international awards, the Editorial Boards of IT International
Journals and Conferences, and other professional bodies and societies. She published 26 Journal and 38 conference papers in International journals and conference proceedings. Her research interests include Software Engineering, Software Testing and Security, DB, E-Learning.

**Nasser Al-Hosni** received his MSc degree in Computer Science in the College of Science at the Sultan Qaboos University in 2013, Muscat, Oman. His research focuses on the proposal of E-Government and Service Oriented Architecture.

**Dr. Naoufel Kraiem**, (PhD from University of Paris VI and HDR from University of Paris1 –Sorbonne - France) is currently Professor of Computer Science in the department of Mathematics and Informatics at the ISI. His research interests include IT adoption and usage Information modelling, Software Engineering, Software product lines and CASE tools. His research work has been supported by funding of the CNRS, INRIA, MRT (Ministry of Research and Technology and Industry) and by the Commission of the European Communities under the ESPRIT Programmers (BUSINESS CLASS) and UNIDO (National Network of Industrial Information Project). He has had several articles published in many journals such as Management Science Information Systems Research Communications of the ACM and IEEE Transactions. He has been invited to present his research in many countries in North America, Europe, Africa and in the Middle East. Pr Naoufel KRAIEM has been member of over 20 program committees. He is in the board of AFCET and he is old consultant at EDS –France from 1990 to 1996. He is currently with the Department of Computer Science at Sultan Qaboos University in Oman.