I. Introduction

With the development of hospital digitalization, the improvement of medical treatment level depends more and more on the efficient management and usage of various kinds of medical information systems. However, current medical information systems are separated from each other and lacking of information sharing and interoperability, which construct barriers for maximal usage of medical resources. Therefore, an integrated large-scale healthcare system which is composed by many interoperable subsystems is needed to adapt to the evolving medical area.

System architecture plays a vital role in the advancement of information systems. System architecture is the overall description of the components and their relationship in the information system; it is the integration of the business logic, information process logic and technical solutions, which is the top model of the information system structure [1].

The purpose of this paper is to provide a comprehensive understanding of system architecture and its role in the information system advancements especially in health information systems. Also, the present architecture models and styles are introduced including the emerging architecture styles. Moreover, a brief description of Service Oriented Architecture (SO implementation in King Fahad Medical City is included.

**Keywords**: System Architecture, SOA, Service Oriented Architecture, AOSD, Aspect Orientation System Development, Healthcare, Clinical System, Information System, medical system, EMR

II. History of System Architecture

In the old days, systems were stand batch data processing that were used in the mainframe. The rapid industry growth and the widespread adoption of personal computer and with the evolution of computer networks the concept of distributed systems has evolved. Client-server architecture has been famous and implemented in different ways and techniques. In the internet age, system architecture has changed drastically; the client software becomes thinner and gradually migrated towards the server.
2.1 What is system Architectures?

There is no universally agreed definition of which aspects constitute a system architecture, and various organizations define it in different ways, including: definitions in Webster's computer industry, it is "the manner in which the components of a computer or computer system are arranged and integrated". [2]

Another definition by ANSI.IEEE 1471-2000 "The fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution." [3]


The Software Engineering Institute of Carnegie Mellon University defines it as "A representation of a system in which there is a mapping of functionality onto hardware and software components, a mapping of the software architecture onto the hardware architecture, and human interaction with these components." [5]

SunTone Architecture Methodology defines it as "A set of structuring principles that enables a system to be comprised of a set of simpler systems each with its own local context that is independent of but not inconsistent with the context of the larger system as a whole". [6]

From the above definitions, it can be concluded that system architecture is an abstract process involving the organization of a system in a structural way that involves, the selection of its elements and their interfaces, behaviors, collaboration and interaction among other elements in a manner with usage, functionality, performance, resilience, reuse, economic and technology constrains and trade-offs The system architecture is a complete methodology that drives and guides the activities of system building and development.

III. System Architecture Importance in Healthcare

Healthcare organization and healthcare system developers affirm that system architecture: [7]

- Applies proven techniques for better systems design and development;
- Increases user's confidence in the developed system;
- Reduces the number of errors produced during the construction phase;
- Reduces the amount of programming efforts to build a coherent system;
- Improves the internal consistency of the system components;
- Serves as integral component of requirements management and traceability and
- Entails easier maintenance, smooth introduction of improvements, updates, errors and corrections.
3.1 System Architecture Models and Styles

Medical systems have become increasingly complex, and the way the system is resembled or built can determine its success or failure. System level architecture specifies how the system functions and subsystems are partitioned, the relationships between the subsystems, what communication exists, modeling these elements outline and the system strategy.

Architectural design is a creative process where one tries to establish a system organization that will satisfy the functional and non-functional system requirements. The activities within the process differ radically depending on the type of the system being developed. The activities include:

- A Static Structural model that shows the sub-systems or components to be developed as separate units;
- A dynamic process model that shows how the system is organized into processes at run-time. This may be different from the static model;
- An interface model that defines the services offered by each sub system through its public interface;
- Relationship models that show relationships such as data flow between subsystems through its public interface; and
- A distribution model that shows how sub-systems may be distributed across computers.

3.2 System Organization Models

The organization of a system reflects the basic strategy that is used to structure a System. The decision about system organization has to be made in the early stages of system design process. Below are three types of organizational styles:

1. The Repository Model: Sub-system making up a system may exchange information so that they can work together effectively.
2. The Client-server Model: The system is organized as a set of services and associated servers and clients that access and use the services.
3. The Layered Model: Organizes a system into layers each of which provides a set of services. An example of layered model is the OSI reference model of network protocols [8].

The Seven Layers of OSI

![Figure 1 OSI Model of Network Protocols](image-url)
Modular Decomposition Styles

After the organization model is determined, the approach of system decomposition into modules should be determined. There is no rigid distinction between system organization and modular decomposition. However, the components in modules are smaller than subsystems.

There are two main strategies that one can use when decomposing a subsystem into modules:

1. Object-oriented decomposition where the system is decomposed into a set of communicating objects.
2. Function-oriented pipelining where the system is decomposed into functional modules that accept input data and transform it into output data.[8]

Control Styles

The models for structuring a system are concerned with how a system is decomposed into sub-systems. To work as a system, sub-systems must be controlled so that their services are delivered to the right place at the right time. There are two generic control styles:

1. Centralized Control: one subsystem has overall responsibility for control and start and stops other subsystems.
2. Event-Based Control: each subsystem can respond to externally generated events. These events might come from other subsystems or from the environment of the system. [8]

The selection of a particular style or model may therefore depend on the nonfunctional system requirements [8][9]:

1. Performance

   The performance requirement is usually measured in terms of response time for a given screen transaction per user. In addition to response time, performance can also be measured in transaction throughput, which is the number of transactions in a given time period, usually one second. The architectural design should take in consideration the performance measurement that allows the designers and developers to complete the system with the requirement.

2. Security

   Security is the ability to ensure that the system cannot be compromised. Security is by far the most difficult to address. Security includes not only issues of confidentiality and integrity, but also relates to Denial-of-Service (DoS) attacks that impact availability. Creating an architecture that is separated into functional
components makes it easier to secure the system because security zones can be built around the components. If a component is compromised, then it is easier to contain the security violation to that component.

3. **Availability**

Availability ensures that a service/resource is always accessible. Reliability can contribute to availability, but availability can be achieved even if components fail. By setting up an environment of redundant components and failover, an individual component can fail and have a negative impact on reliability, but the service is still available due to the redundancy.

4. **Maintainability**

Maintainability is the ability to correct flaws in the existing functionality without impacting other components of the system. When creating an architecture and design, the following should be considered to enhance the maintainability of a system: low coupling, modularity, and documentation.

IV. **Emerging Architectural Styles**

With the rapid development of information technology, computer sciences and growing business demand a new kind of information systems where new development techniques can also be introduced. The development and design of those systems requires innovative architectural styles that support the new demands. Among them [8]:

- Service Oriented Architecture (SOA)
- Aspect Oriented Software Development (AOSD)

4.1 **Service Oriented Architecture (SOA)**

Service-oriented architecture is a hot topic in software development field to reduce the dependencies between different information systems, and it is an evolution of distributed computing based on the request/reply design paradigm for synchronous and asynchronous applications.

4.1.1 **Service Oriented Architecture Definitions**

SOA acronym becomes widely used, but there is not a lot of precision in the way that it is used. Service-oriented architecture’ refers to systems structured as networks of loosely coupled, communicating services. [10]

- Service Oriented architecture (SOA) is defined in the OASIS reference model as "a paradigm for the organizing and utilizing distributed capabilities that may be include the control of different ownership domains. [11]
The World Wide Web Consortium (W3C) for example refers to SOA as 'A set of components which can be invoked, and whose interface descriptions can be published and discovered'.[12]

SOA is the architectural style that supports loosely coupled services to enable business flexibility in an interoperable, technology-agnostic manner. SOA consists of a composite set of business-aligned services that support a flexible and dynamically re-configurable end-to-end business processes realization using interface-based service descriptions. [13]
4.1.2 SOA Platform

Development of SOA platform entails two major tasks [8][14] [15]:

1. Service Infrastructure

Establishing service infrastructure (such as, core services, management and security), which can be viewed as the combination of the interface layer and the attached execution environments. It is used to run and manage SOA applications. Service Infrastructure must support all the relevant standards and required run time containers such as WSDL, UDDI, SOAP and WS-I.

![Service Infrastructure Diagram](image)

**Figure 3** Service Infrastructure

2. Service Architecture creates composite application or services based on the service infrastructure. Service architecture can provide a business rules engine that allows business rules to be incorporated in a service or across services. The service architecture also provides a service management infrastructure that manages services and activities like auditing, billing, and logging. In addition, the architecture offers enterprises the flexibility of having agile business processes, better addresses the regulatory requirements, and changes individual services without affecting other services.

4.1.3 SOA Main Characteristics

Service-oriented architectures have the following key characteristics and standards [8][15]:

1. SOAP is a message interchange standard that supports the communication between services. It defines the essential and optional components of messages passed between services.

2. WDSL the Web Service Definition Language (WSDL) standard defines the way in which service providers should define the interface to those services. Essentially, it allows the interface of service (the service operations, parameters and their types) and it’s binding to be defined in a standard way. Example platform-independent XML.

3. UDDI the UDDI (Universal description, discovery and integration) standard defines the components of a service specification that may be used to discover the existence of a service. These include information about the service provider; the service provided the location of service description (usually expressed in WSDL) and information about business relationship. UDDI registries enable potential users of a service to discover what services are available.

4. WS-BPEL is a standard workflow language that is used to define process programs involving several different services.

Existing mission-critical systems in enterprises address advanced requirements such as security, reliability, and transactions which is called in SOA terminology as quality of service (QOS). Numerous specifications related to QoS are being worked out in standards bodies like the World Wide Web Consortium (W3C) and the Organization for the Advancement of Structured Information Standards (OASIS).

Sections below discuss some of the QoS artifacts and related standards [8][15].

Security

The Web Services Security specification addresses message security. This specification focuses on credential exchange, message integrity, and message confidentiality.

Reliability

In a typical SOA environment, several documents are exchanged between service requesters and service providers. Delivery of messages with characteristics like once-and-only-once delivery, at-most-once delivery, duplicate message elimination, guaranteed message delivery, and acknowledgment become important in mission-critical systems using service architecture. WS-Reliability and WS-Reliable Messaging are two standards that address the issues of reliable messaging.
Policy

Service providers sometimes require service consumers to communicate with certain policies. As an example, a service provider may require a Kerberos security token for accessing the service. These requirements are defined as policy assertions. A policy may consist of multiple assertions. WS-Policy standardizes how policies are to be communicated between service consumers and service providers.

Orchestration

As enterprises embark on service architecture, services can be used to integrate silos of data, applications, and components. Integrating applications means that the process requirements, such as asynchronous communication, parallel processing, data transformation, and compensation, must be standardized. BPEL4WS or WSBPEL (Web Services Business Process Execution Language) is an OASIS specification that addresses service orchestration.

Management

As the number of services and business processes exposed as services grow in the enterprise, a management infrastructure that lets the system administrators manage the services running in a heterogeneous environment becomes important. Web Services for Distributed Management (WSDM) will specify that any service implemented according to WSDM will be manageable by a WSDM-compliant management solution.

4.1.4 SOA Impact on Business

- Responds to business changes with agility, leveraging existing investments in applications and application infrastructures.
- SOA loosely coupled nature allows the organization to plug in new services or upgrade existing services in a granular fashion to address new business needs utilizing existing infrastructure.
- Safeguards existing IT infrastructure investments.
- Ensures Better reusability of existing infrastructure and its investments
- Guarantees high flexibility in building application and business processes in an agile manner by leveraging existing application infrastructure to compose new services.
- Boosts return on IT investment through reduced costs, elimination of redundancy, and improved productivity, among other.

4.1.5 SOA Technologies and Platform

J2EE and .Net platforms are the dominant development platforms for SOA Applications. SOA is not by any means limited to these platforms. Platforms such as J2EE not only provide the framework for developers to naturally participate in the SOA, but also, by
their inherent nature, bring a mature and proven infrastructure for scalability, reliability, availability, and performance to the SOA world.

V. SOA in Healthcare

The rapid growth of information technology and its adoption in the healthcare industry has caused healthcare organizations to have non-interoperable and legacy systems that need to work together. This led to integration burden and complications. Also, the healthcare organizations are challenged to manage a growing portfolio of systems. The use of SOA can improve the delivery of information and make it available horizontally and vertically in reasonable cost and minimized security and deployment risks without the need to re-engineer the existing systems. This means existing system capabilities increase in value as they are packaged and shared as services. [16].

SOA defines a service as an independent unit of work that is self contained and has well-defined, understood capabilities. A unit of work may be an entire process, a function supporting a process, or a step of a business process. Each system function may be separated into tasks to further increase reuse opportunities for services. For example, the function "register patient" may be separated into the tasks "find and view patient record," "create and update patient record," "verify insurance eligibility," "document history" (new or update), and other business activities completed during the registration process. This granularity allows other services and applications to use parts of the "register patient" function. The task "find and view patient record" may be used by most of the organization, whereas the task "create and update patient record" may be used only by the admission and eligibility staff [16]. Figure 4 presents a conceptual view of the "register patient" services function:
5.1 Extending Electronic Medical Records through SOA

Many healthcare organizations around the world are planning or putting in place Electronic Medical Records (EMR) systems to automate the collection, distribution, and validation of patient medical records. SOA can support the following scenarios [16]:

1. An SOA-based EMR can readily support many forms of user interface because the core data and business logic functions are loosely coupled from the presentation.

2. An SOA-based EMR is constructed as a suite of compassable software services for data and business rules. Workflows can be more readily customized to support individual organizational or departmental needs without having to resort to a "best-of-breed" deployment where individual departments have nearly duplicate application stacks from different vendors since one vendor supports a particular department’s workflows incrementally better.

3. The SOA architecture allows for entire functions of an EMR system’s or hospital’s processes to be outsourced and hosted in a shared data center and consumed as a utility. The key advantage to this is the cost of implementing and sustaining this functionality. More often than not, acquiring software functionality through the outsourced, hosted utility model (sometimes referred to as utility computing or application service provider) can be done for a materially lower overall total cost. When using SOA techniques and technology, a healthcare IT organization can readily integrate internally hosted systems and technology directly alongside outsourced ones. Figure 5 describes this architecture.

![Figure 5 Architecture for Integrating Hosted and Outsource Applications](image-url)
VI. SOA in KFMC

The Adoption of Service-Oriented Architecture (SOA) has been promoted in King Fahad Medical City as a technology that enhances systems agility, interoperability between applications, deployment flexibility, reusability, and generate sustained competitive advantage. Integration and in-house development team in KFMC has recognized Service-oriented Architecture (SOA) as an approach for building systems that enhances IT's ability to efficiently and effectively respond to the rapidly changing business environment and enables organizations to respond to these changes in a timely manner. Also, SOA provides methods for systems development and integration where systems group functionality around business processes and package these as an interoperable service with the emphasis of data flow horizontally and vertically through KFMC application infrastructure.

KFMC has learnt through literature review on SOA adoption and implementation that ill-planned IT systems and services might yield disruptive services and business loss and big-bang theory is not an option. The implementation efforts were paced in phased to align with organization's overall ability to absorb change through development of realistic, achievable plan that includes sequencing deployment. Moreover, the adoption and the transition involve uncoupling preexisting systems, creating modular components and loosely recouping these components via SOA common standard. The Project of adopting SOA is still now in its early phases and expected to be full SOA compliant by the end of year 2011. It is expected to have a solid technical team that is specialized in SOA architecture, designing and development.

KFMC development and integration team is strongly recommended to adopt a full software development cycle including requirements gathering and analysis and proven test methodologies. Also, they are recommended to have an automated system to mechanize the process of service-oriented architecture.

VII. Aspect Oriented Software Development (AOSD)

Definition

AOSD is an emerging approach to software development that is intended to address the problem of changing requirements an system abstraction to make systems easier to maintain and reuse. AOSD is based on a new type of abstraction called aspect intended to isolate secondary or supporting functions from the main program's business logic. AOSD allows multiple concerns to be expressed separately and automatically unified into working systems [8].
Key Benefits

The motivation for aspect-oriented programming approaches stem from the problems caused by code scattering and tangling. The purpose of Aspect-Oriented Software Development is to provide systematic means to modularize crosscutting concerns. Aspect-oriented software development considers that code scattering and tangling are the symptoms of crosscutting concerns. Crosscutting concerns cannot be modularized using the decomposition mechanisms of the language (object or procedures) because they inherently follow different decomposition rules [17].

Some companies such as IBM are starting to use AOSD in their software production process. Support separation of concerns into independent elements rather than including different concerns in the same logical abstraction is AQ good software engineering practice for reusing and modifying elements/concerns indecently [8].

Important characteristics

The focus of Aspect-Oriented Software Development (AOSD) is in the investigation and implementation of new structures for software modularity that provide support for explicit abstractions to modularize concerns. [8][17]

1. Includes definition of where the aspects should be included in a system.
2. Includes definition of the code implementing cross cutting concern.
3. Enables the core program to create and augment new systems.

AOSD Adoption

AOSD has been adopted by many leading companies including programming languages such as [17]:

- BM Websphere Application Server. Websphere uses Aspect internally to isolate features of the different editions.
- JBoss Application Server (JBoss AS) is a free, open-source java application server that supports Java EE. The core of JBoss AS is integrated with the JBoss AOP aspect-oriented programming language.
- Oracle TopLink is a Java object-to-relational persistence framework that is integrated with the Spring Application Server. TopLink achieves high levels of persistence transparency using Spring AOP.
- SAP
- Sun Microsystems uses Aspect to streamline mobile application development for the Java ME platform. Aspects are used to simplify the development of mobile applications for deployment to different operator decks and different mobile gaming community interfaces.
- Siemens Soarian is a health information management system that supports seamless access to patient medical records and the definition of workflows for
health provider organizations. Soarian uses Aspect to integrate crosscutting features such as tracing, auditing and performance monitoring in the context of an agile development process.

- .NET 3.5 supports Aspect Oriented concepts through the Unity container.

VIII. Conclusion

System architecture can be viewed as the methodology and the guiding structure that defines how the system elements interact between each other and with the element of external system. System architecture is important since it allows for expansion without the need to re-engineer the whole infrastructure, allows data flow horizontally and vertically, and preserves the investments. System architecture has different model and styles which can be determined based on business demand. Some emerging architecture styles are now in the market such as service oriented architecture (SOA) and aspect oriented architecture (AOSD). Those styles has changed the medical systems development dramatically in innovative ways for better resource utilization, service delivery and better quality care.

IX. References


2. www.webster.com, 24 Nov 2010


12. http://www.w3.org/


