Airport Enterprise Service Bus with Self-Healing Architecture (AESB-SH)

Issam Hamad Al Hadid, Isra University, Jordan

ABSTRACT

Airports need to adapt new technologies to react effectively and quickly to customers’ needs and to provide a better service such as the electronic ticket. In addition to the challenges of the ability to respond to the growing requirements of the automatic information interchange between the different systems to ensure safe and efficient airport operations. This paper provides an architecture based on the Service Oriented Architecture (SOA) that improves the information accessibility and sharing across the different Airport’s departments, integrates the existing legacy systems with other applications, and improves and maximizes the system’s reliability, adaptability, robustness, and availability using the Self-Healing Agent.

Keywords: Airport and Aviation IS, Airport Management System, Enterprise Service Bus, Self-Healing, Service Oriented Architecture (SOA)

1. INTRODUCTION

Most of the challenges in the Airport systems today lie in the ability to respond to the growing requirements of the automatic information interchange between the different departments including operational, statistical, aviation and financial information. In addition, the integration with the existing legacy systems ensures safe and efficient airport operations. All the operations in the airport are driven by the exchanged information; Airport business units create information, transform information, distribute information, and take action on received information. Airports’ systems are developed by different vendors and were not designed to be interoperable, which makes systems integration a very complicated and not easy to be implemented. Service Oriented Architecture (SOA) provides the ability to address the distributed computing requirements; protocol independent, loosely coupled, reusability and standard based (Papazoglou & Heuvel, 2007). It is based on the Web Services; distributed, loosely coupled, reusable software components that encapsulate a discrete functionality and can be accessed using standard internet and XML-based protocols (Sommerville, 2007). SOA encourages a lot of businesses to move toward the adapting the SOA architecture to enable the response to change faster and to cut the cost of replacing the legacy systems that they have and integrate with the new systems so all the information can be accessed and shared by all the systems (Keen
et al., 2004; Minoli, 2008), accordingly; SOA will provide a guideline for airport Information systems architecture design, development and integration. The functionality is provided by the integration platform Enterprise Service Bus (ESB). It is based on the SOA that utilizes Web Service standards to support a variety of communication patterns over multiple transport protocols to connect different applications and technologies (Papazoglou & Heuvel, 2007). In addition, the features of loosely coupling and breaking up the integration logic into separate parts can be easily managed (Keen et al., 2004).

ESB provides architecture based on the SOA that improves the information accessibility and sharing across the different Airport’s departments, furthermore, it provides a component interface to existing legacy system so it can be integrated with other applications, accessed over the web, and support the reusability of the legacy systems. Also, the Self-Healing Agents which will improve and maximize the system’s reliability, adaptability, robustness and availability.

2. RELATED WORK

There are many different information technology systems that are used in the aviation industry (Abu-Taieh, 2009), shown in Table 1. Many Airports and Air Traffic Control (ATC) Units have moved toward the adapting of new open systems (Goold, n. d.). The information can be accessed, shared and flow across the different hardware systems, operating systems, networks and airport management systems, because there must be an integration framework for defining the information integration requirements, and designing the systems integration architecture to address the loosely coupled systems, standards-based interfaces, and protocol independent distributed computing. This manages information elements with a defined process and provides the ability to upgrade, replace or move systems or components without having to modify code and disrupt execution of the existing applications (Cheng, 2001). Service Oriented Architecture (SOA) provides the ability to address the distributed computing requirements; protocol independent, loosely coupled, reusability and standard based (Papazoglou & Heuvel, 2007). It is based on Web Services; distributed, loosely coupled, reusable software components that encapsulate a discrete functionality and can be accessed using standard internet and XML-based protocols (Sommerville, 2007). SOA provides flexible architecture that unifies the business process by modularizing applications into services which satisfy the addressed requirements for the Airport and aviation systems integration architecture. The functionality of the integration architecture must also support a variety of communication patterns over multiple transport protocols, this requirement is addressed by the integration platform Enterprise Service Bus (ESB) which is based on the SOA. It utilizes Web Service standards to support a variety of communications patterns over multiple transport protocols to connect different applications and technologies (Papazoglou & Heuvel, 2007). It also, provides the integration logic between the service consumer and provider which is used to transform messages, route requests and convert transport protocols between the two parties (Keen et al., 2004; Chappellm, 2005). The configuration and orchestration of the services in unified and clearly defined processes are provided by the Service Orchestration using the Business Process Execution Language (BPEL), this allows the business operations to have the ability to respond to the underlying business needs via the different components invocations either in an event-driven or asynchronous fashion to fulfill a complex business process. As well, ESB improves the ability to upgrade, move or replace applications or services without having to modify code and disrupt existing ESB applications; due to the abstraction of the physical destination and connection information provided by the End points, they allow services to communicate using logical connection names which will be mapped to actual physical network destination at runtime (Papazoglou & Heuvel, 2007; Chappell, 2004). Beside, the Self-Healing Agent that will be added to the