Comparative Study on Service-Oriented Architecture and Event-Driven Architecture

Chatri Pienwittayasakul and Yan Liu
School of Software Engineering
Tongji University, Shanghai, P.R. China
chatri.pi@gmail.com, yanliu.sse@tongji.edu.cn

ABSTRACT

Over the years many different architecture styles and concepts have evolved. Two of them are Service-Oriented architecture (SOA) and Event-Driven architecture (EDA). Both styles are revolutionary in the way they try to link business with information technology. They can provide great benefits for the business organization if they are used in the right occasion for the right purpose. However, there are some confusion about the relation between these two architectures. There is a lot of debate on exactly how Event-Driven architecture will blend in with Service-Oriented architecture. Before adopting Service-Oriented architecture and Event-Driven Architecture for architectural design of enterprise system, it would be better to obviously understand the similarity and distinct between Service-Oriented architecture and Event-Driven Architecture. The study is trying to describe two topics Service-Oriented architecture and Event-Driven architecture in several aspects (characteristics, concept, principles, data, and implementation components). These aspects are studied and analyzed by using existing information. From the comparative study, it could be summarized that Event-Driven architecture is not an implementation style of Service-Oriented architecture but they are peers and complements within the business context and IT context.

KEYWORDS

Architecture; Service-Oriented Architecture; Event-Driven Architecture; Event; Service

1 INTRODUCTION

Information technology can be referring to any field that relate to computer technology including software development, hardware maintenance and networking and infrastructure solution. Most modern business enterprises and organizations all over the world highly depend on information system. Information technology has become unavoidable to align with business. So, it is beneficial to understand how the role of information systems has evolved in the organization.

In the enterprise architecture field, the new standard-based architecture paradigm promises great advances in interoperability among previous incompatible software applications. It has the potentials to deliver gains in agility and cost control. One of them is the concept of Service-Oriented architecture (SOA). It has been raised and observed by the many professional for several years. Another one is Event-Driven architecture (EDA). This architecture increases system's awareness and intelligent responses to relevant events.

EDA has been introduced for many years. It is not a new architecture style. EDA has been recovered for several years according to the availability of new technology and SOA. Today EDA is mentioned when discussing SOA more often. However, the implicit connection between EDA and SOA has not been obvious even for a lot of experienced architects and developers. There are different opinions about the way SOA and EDA function together. Some consider EDA to be an extension of SOA or complement to SOA but others consider EDA to be a part of the SOA.

The objective of this paper is to study Service-Oriented architecture (SOA), Event-Driven architecture (EDA), and their relation. The scope of study will cover mainly in conceptual level which plans to find and discuss similarities and differences between SOA and EDA. It also identifies characteristics and interaction points where both SOA and EDA are applied to achieve their promised benefits.

2 SERVICE-ORIENTED ARCHITECTURE (SOA)

2.1 Background

For the term SOA, it is quite hard to find an exact definition. The reason is not because there is no definition but there are many different definitions. Some selected published definitions are listed below in order to give an idea of how they are similar and different.

- A software design and software architecture design pattern based on discrete pieces of software providing application functionality as services to other applications. This is known as service-orientation. It is independent of any vendor, product or technology [1].
- A term that represents a model in which automation logic is decomposed to smaller, distinct unit of logic. Collectively, these units comprise a larger piece of business automation logic. Individually, these units can be distributed [2].
- A conceptual business architecture where business functionality, or application logic, is made available to SOA users, or consumers, as shared, reusable services on an IT network. "Services" in an SOA are modules of business or application functionality with exposed interfaces, and are invoked by messages" [3].

These definitions have an intention to demonstrate SOA in many different points of view which neither of them is right nor wrong by itself.

2.2 Characteristic of SOA

Nowadays, an important factor for the business to be able to survive is the use of different kind of technology for their daily operation and long term strategy. Organizations are forced to broaden their connectivity and increase their revenues. They also focus on innovation about reconstructing application to reduce cost and increase flexibility. It is almost impossible to complete long-term planning while markets keep changing. So, an ability to plan continuously and quickly is crucial. SOA can serve this demand by providing business agility.

SOA provide the ways to plan, design, and deliver IT functionality as modular business services in order to meet some specific business requirements. All organizations can directly have advantages from SOA. For the business, SOA means real business agility, increase in customer satisfaction, decrease in business cost and ease of partnering. For IT organization, SOA means decrease in IT cost, greater asset reuse, higher quality applications, overall faster application development, enhancement and modification.

SOA also provides the capability to loosen tight dependency of applications, trading partners, and organizations. Furthermore, services can be composed to form a process in order to provide greater value. SOA aims to eventually realize business agility for organizations. When discussing SOA, the word "agility" is often mentioned. This word can be described into two forms. Frist, it is ability to faster run business processes or launch new processes. Second, it is an ability to reduce cost and adapt business process according to the change in market demands.

2.3 Concept

A goal of SOA is to achieve loose coupling among a set of services in relation. Service provider and service consumer are two important roles played by software agent and units of organization. A service is a unit of work done by a service provider to achieve expected results for a service consumer. SOA use synchronous communication (request/response) and implemented in one-to-one relationship. A service consumer needs to wait until the completion of the operations on provider side after invokes a service provider through the network [4].

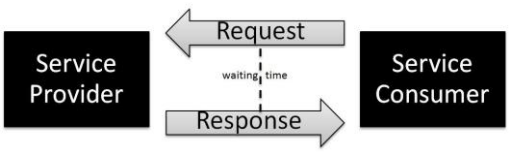

Figure 1. Request/response mechanism in SOA

Service-oriented architecture consists of business services. These services are flexible and dynamically configurable. The concept of service-oriented is illustrated by a well-known triangle. There are three important steps for service interaction. First is to register the service to a registry. Second is to locate the service. Third is to make service calls and exchange messages. Service providers offer their services through its interface for service consumers. This interface describes the contract between the service provider and service consumer. It also defines information about available services and agreement for using the interface [5]. All available services are contained in a network-based directory called registry. It is an entity that accepts and stores contracts from service providers and provides those contracts to interested service consumers [6].

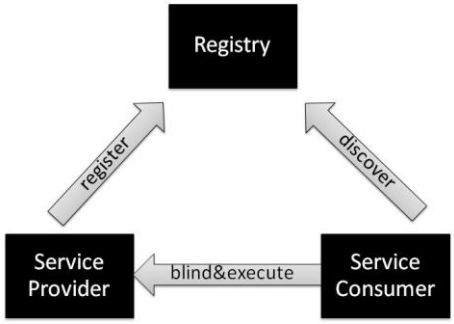

Figure 2. Concept of Service-Orientation

2.4 Principle

Below are the eight service-oriented design principles together with their official definitions.

- Standardized service contract: Services within the same service inventory are in compliance with the same contract design standard.
- Service loose coupling: Services contracts impose low consumer coupling requirements and are themselves decoupled from their surrounding environments.
- Service abstraction: Services contract only contain essential information and information about services is limited to what is published in service contract.
- Service reusability: Service contains and expresses agnostic logic and can be positioned as reusable enterprise resources.
- Service autonomy: Services exercise a high level of control over their underlying run-time execution environment.
- Service stateless: Services minimize resource consumption by deferring the management of state information when necessary.
- Service discoverability: Services are supplemented with communicative metadata by which they can be effectively discovered and interpreted.
- Service Composability: Service is effective composition participants, regardless of the size and complexity of the composition.

2.5 Data

One of the most valuable assets of an organization is data. The data are typically stored separately in different data silos. However, the data sources in an organization are not designed to be used globally by all information systems. Thought, it can provide relevant information to a particular business unit, it also creates inconsistencies in data at the enterprise level. This challenge can be solved by the solutions provided by SOA. SOA represent a new way of thinking about data. It includes some guidelines and patterns to manage enterprise‘s data properly. There are three important topics when discussing information management in SOA.

2.5.1 Data integration

It involves combining data residing in different sources and providing user with a unified view of the data in order to make the data more reliable and consistent. Applying SOA enables the business to access, manipulate, and share data across the organization in a consistent and technology independent way which helpful when supporting the business decision making. In SOA data are delivered as a service. This can loosen the tight connections between data and information systems so that data can be controllable and

sharable across the enterprises. Business rules are defined in order to make the data consistent. These rules are about how the data is structured, accessed and used. This common understanding of the data is important and must be extend across business units and regions.

2.5.2 Data semantics

It is the meaning of data. Semantics of data is used to ensure that everyone in the organization has a common understanding about business information and business rules. One of the main objectives of SOA is to prevent business from running into chaos by providing accurate business information to everyone in the organization. One major step is to ensure that each component of data can be used independently from its current technology and implementation. In SOA, data is considered as a reusable resource which is called "information as a service" [7].

2.5.3 Meta data

It is data about data. It contains definitions, relations and other characteristics used to authorize access and use of company's data. Business services need to be able to access metadata in order to consume and deliver the required data. Metadata is stored in the metadata repository. It contains definitions of business data and rules for mapping data to their actual physical location in the information system. This repository is in a technical layer between the actual data stores and the business services. The purpose of the metadata repository is to ensure that the data is in the right structure and quality before it is consumed by a business service. The metadata repository also ensures that data from different sources can be linked together correctly [7].

2.6 Implementation components of SOA

2.6.1 Business Services

Business Services are the exposed part of the business processes and the business functionality that is accessed by and provides predefined values to the requestor.

2.6.2 Application Services

The Application Services domain comprises all of the components that are necessary to build a composite information system. This domain includes components to support user access, business function, common services, user interaction, business process choreography, and information management.

2.6.3 Common Services

The Common Services components enable personalization of the delivery and individual processing, as well as other utilities such as reporting. These services are used to define and execute conditional flows mainly across services of logical business function but also to other service.

2.6.4 Information Management Services

The Information Management Services provide a uniform way of representing, accessing, maintaining, managing, analysis, and integrating data and content across heterogeneous sources of information.

2.6.5 Enterprise Service Bus

The Enterprise Service Bus is an intermediation layer that interconnects all of the services, enabling all of the (loose) coupling characteristics.

2.6.6 Infrastructure Services

Infrastructure services is a set of platform independent services that enables all of the other services domain components to be installed, executed, and controlled on a concrete infrastructure combination of operating systems and network and hardware systems [8].

2.7 Enterprise Service Bus (ESB)

Enterprise Service Bus acts as an intermediary, supplies loosely coupled connectivity between the participants in service interactions. ESB provides the backbone of an SOA. ESB enables the connection between software that run on different platforms. The following figure illustrates the logical relationship between service requesters, service providers, and the ESB. Service requesters and providers interact by exchanging messages [9].

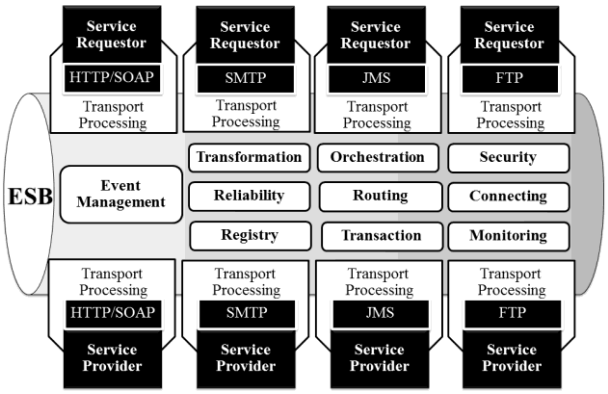

Figure 3. ESB architecture

ESB acts as the intelligent layer for connecting information systems, various kinds of data, and other services that are commonly distributed throughout an enterprise IT environment. It combines its core synchronous and asynchronous messaging backbones with capabilities to intelligently transform the data and route the message. It also ensures that messages are passed reliably. ESB enables the intelligent processing of service requests, service responses, events, and messages which is much more interesting when discussing EDA [8].

2.8 Summary of SOA

SOA is an architectural paradigm for dealing with business processes distributed over a large landscape of existing and new heterogeneous systems environment from different partners. The keys technical concepts of SOA are services, interoperability and loss coupling. The key ingredients of SOA are infrastructure, architecture and process. The key success factors for SOA are understanding, governance and management support. SOA is neither a specific technology nor a silver bullet. There are some places where SOA is appropriate and some places where it is not. Web service is one possible way of realizing the infrastructure aspect of SOA. Using Web service is often recommended because it seems to be becoming established as the standard technology.

3 EVENT-DRIVEN ARCHITECTURE (EDA)

3.1 Background

Event-Driven software architecture and design patterns have been around and implemented for many years. Due to the entry of SOA for several years, Event-Driven architecture has been revived.

3.2 Characteristic of EDA

EDA have similar objectives as SOA because they try to achieve and increase agility and flexibility within organizations. The major strategy of Event-Driven architecture is to maintain competitive flexibility by structuring itself for immediate response to business events in its surrounding environments, modifying its organization and operations in real-time to give customers exactly what they want. The common advantages that EDA bring to organizations are listed below.

- EDA gather main user, customer and all operational unit in marketplace together in real-time without barriers of distance or technology.
- EDA provide each employee, partner and customer with a comprehensive set of real-time, constantly upgrade and customer designed information by using proactive publish/subscribe instead of passive request/response paradigm.
- EDA promote value-added and active information sharing.
- EDA infrastructures allow the system to be able to learn from its mistakes. It is impossible to sweep anything under the rug. The company will be aware of what is going wrong at the moment it starts to go wrong.

3.3 Concept

An event is a notable thing which happens inside or outside your business, which disseminates immediately to all interested parties (human or machine). The interested parties evaluate the events, and optionally take action. The action may include the invocation of a service, the triggering of a business process and/or further information publication/syndication.

EDA is based on asynchronous publish/subscribe pattern, where the publisher is completely unaware of the subscriber and vice versa. They are loosely coupled in the sense that they only share the semantics of the message. An event driven architecture is extremely loosely coupled, and highly distributed by nature. The producer of the event only knows the event transpired. The producer has no knowledge of the subsequent processes of event processing, or the interested parties. The traceability of an event through a dynamic multipath event network can be difficult. Thus, event-driven architectures are best used for asynchronous flows of work and information.

Business event definition process is summarized in several steps. The event definition process should start by recognizing the business events that your business is planned to react upon. Next step is to define the process that detects the event. Event shall be designed in such a way that it is not only fit to current requirements but also future requirements by focusing on the characteristics of the event itself [10].

Each event occurrence has an event header and event body. The event header contains elements describing the event occurrence, such as the event specification ID, event type, event name, event timestamp, event occurrence number and event creator. These elements are consistent across event specification [11].

The event body describes what happened. For example, if a retailer specified a low inventory threshold event, the event body would contain the information to communicate which product fell below the allowable threshold. The event body must be fully described because all interested party can use the information without having to go back to the source system. In order to ensure that events are understood by all consumers, a clear business lexicon or ontology should be used [11].

3.4 Principle

The summary of the main principles are listed below

- Decoupled: The event producer only knows the event created. This decouples application integrations and business processes. Each process can be modified without causing side effects on the others. [12].
- Reusability: Events must be designed in a generic way which is able to be reused [10].
- Real-time notification: The technical infrastructure must support real-time creation and delivery of events. The human infrastructure must support real-time transformation of information first into knowledge and then into wisdom [13].
- Publish/Subscribe: Notifications are pushed by the event producer, not pulled by the event consumer.
- Immediate response: The arrival of a notification causes the event consumer to react immediately.
- Freely react: The consumer contains the logic that determines how it will respond or react [12].
- Statelessness of event processing components: The event processing components are stateless because the state is defined with in an event.

3.5 Data

In Event-Driven architecture, information is distributed in real-time to involved parties in an organization. After information is delivered by using publish/subscribe communication mechanism, the enrichment loop begin. It insert the data into applications that analyze and refine the data then send it to all interested parties.

Data distribution and loose coupling also put some requirements on the data in an EDA. Loose

coupling means independency where components do not rely on each other. Each loosely coupled environment maintains its own copy of the data. This leads to data redundancy. However, maintaining redundancy across the decoupling borders (synchronization) can make the loose coupling more robust [10].

EDA also requires shared semantic in order to connect all information systems together. It should be obvious to anyone that analysis of data semantics will always be the first activity of any integration project.

In an EDA, business event is represented in a standardized format base on agreed semantics. This format and semantics are defined in the enterprises metadata model. It is sometimes called Canonical Data Model (CMD). Enterprise Service Bus is responsible for transformation of data between the local and canonical format so everyone can use their own model, formats and semantics.

3.6 Implementation components of EDA

The implementation components of EDA can be divided into five categories [11].

3.6.1 Metadata

A good Event-Driven architecture has strong metadata architecture. Event metadata includes event specifications and event processing rules. Event specifications must be made available to event generators, event format transformers, event processing engines and subscribers.

3.6.2 Event Processing

The cores of event processing are the engine and the event occurrence data. Simple event engines are often homegrown. Complex event engines should be acquired from a CEP engine provider. Event occurrence data is typically persisted and retained for audit and trend analysis.

3.6.3 Event Tooling

Event development tools are required to define event specifications and processing rules and to manage subscriptions. Event management tools provide administration and monitoring of the event processing infrastructure, monitoring of event flows and visibility into event generation and processing statistics.

3.6.4 Enterprise Integration

An enterprise integration infrastructure plays a large role in EDA to filter, route, transform, transport, invocate services and invocate business processes.

3.6.5 Sources and Targets

These are the resources of the enterprise applications, services, business processes, data stores, people and automated agents that generate events and/or perform an Event-Driven action.

3.7 Complex Event Processing (CEP)

CEP deals with evaluating a confluence of events and then taking action. It may cross event types and occur over a long period of time. The event correlation may be casual, temporal, or spatial. CEP requires the employment of sophisticated event interpreters, event pattern definition and matching and correlation techniques. CEP is commonly used to detect and respond to business anomalies, threats, and opportunities.

A CEP system has some capabilities to organize random and unrelated events, find trends and to predict outcomes. After that action can be taken to prevent an unexpected outcome from occurring. The patterns can also be analyzed to improve the underlying processes and information systems to prevent future mistakes.

3.8 Summary of EDA

Providing proactive and adaptive environment to organization with real-time communication between all interested parties is the key of EDA. EDA requires some specific infrastructure service such as EAI, CEP and metadata repository. It supports decoupled architecture because services are unaware of existence of each other. There is still an obvious gap in supporting design

principles, standards, infrastructures, and design tools even though EDA has been in use for many years.

4 Comparative Analysis

The comparative analysis describes the differences and similarities between the two architecture styles and how they interact and affect each other in a common environment. SOA and EDA are compared in different views and aspects. This comparative analysis is divided into four different categories. Those are business, information system, integration and information.

4.1 Business

SOA and EDA use different means in order to align the business. SOA use business service while EDA use business events to bridge the gap between business and information technology. For business, SOA and EDA means increase in customer satisfaction, real business agility, faster time to market, better business intelligence and lower business cost at the end.

Table 1. Comparative analysis-business

SOA	EDA
•SOA focus on decomposition of business functions which can be reused •SOA is an architecture style that recognizes services (functionality representing process steps) •SOA facilitate business process automation and integration •SOA promote reusability of functions (reactive)	•EDA focus on identifying business events. (an event is a change with in the state of an enterprise) •EDA is an architectural style that recognizes events (message representing process states) that your business is planned to react upon •EDA promote information sharing and use of business intelligence (proactive)

4.2 Information System

Table 2. Comparative analysis-information system

SOA	EDA
•Applies incremental development and maintenance of large distributed applications •Systems are modeled as Service Providers and Service Consumers	•Applies incremental development and maintenance of large distributed applications •An approach for designing and building systems in which events trigger messages to be sent between independent software modules that are completely unaware of each other

4.3 Integration

Table 3. Comparative analysis-integration

SOA	EDA
•A SOA-based application consists of business components that supply services and other programs that act as clients or "consumers" of those services •Promotes integration by introducing standards and separating steps of wrapping, layering and composing •Request/Response pattern •Can use both synchronous and asynchronous patterns •Loose coupling is provided because the service providers and service consumers use the service definition as a communication and interaction contract •Services are invoked independently of their technology and location • One specific service is invoked by one consumer at a time (one-to-one communication)	•Information system or services broadcasts state changes using events. Other systems that are interested in these events can subscribe to these events and choose to respond to it •Suggest asynchronous pattern •Decoupling is provided since event publishers are not aware of the existence of event subscribers •Publish/Subscribe messaging where one specific event can impact many subscribers (many-to many communication)

4.4 Information

Table 4. Comparative analysis-information

SOA	EDA
•Includes structured data •Use of Canonical Data Model •Data redundancy is not promoted nor declined •Information is requested by the ones who are interested in that information •Promotes reuse and share of data in multiple applications and for multiple end user access channels •Information as a service is an architectural approach that loosens the tight connections between data and applications so that data can be controlled and shared across the enterprise •Master data provides"information services" with data	•Includes structured data •Use of Canonical Data Model •Data redundancy is a necessity to increase decoupling •Information is distriuted in real-time to the ones who are interested in that information •Promotes instant identification and responding to the events/information that drive the business •Uses the power of information to drive the development

5 CONCLUSIONS

It became evident, from studying the literature and conducting a comparative analysis, that there are several areas where EDA and SOA are interacting. One or many services can be invoked by the occurrence of an event. These services can perform simple functions or entire business processes. This is the first interaction and

commonly referred as a style of SOA (Event-Driven SOA). Another interaction is that an event can be generated by a service. This event can signify opportunity, threshold or problem. An event is disseminated to all interested parties once it's generated. Events are evaluated by the parties. The parties optionally take action. The action can be publication of information, service invocation or triggering of business process. This style services are just event sources in Event-Driven environment. It is the author's opinion that SOA and EDA both are architectural styles. EDA is not just an implementation style of SOA. Service-Oriented architecture and Event-Driven architecture are peers and complements within business context and IT context.

6 ACKNOWLEDGMENT

This work was financially supported by the Science and Technology Commission of Shanghai Municipality (12dz1507000).

7 REFERENCES

[1] Wikipedia contributors, "Service-oriented architecture," Wikipedia, The Free Encyclopedia, < http://en.wikipedia.org/w/index.php?title=Service-oriented_architecture&oldid=600126256 >, March 2014

[2] Thomas, E., "Service-Oriented Architecture (SOA): Concepts, Technology, and Design," Aug 12, 2005

[3] Marks, E.A., Bell, M., "Service Oriented Architecture A planning and implementation guide for business and technology," John Wiley and Sons, Inc., 2006

[4] Marechaux, J,L., "Combining Service-Oriented Architecture and Event-Driven Architecture using an Enterprise Service Bus," IBM Software Group, 2006

[5] Allen, P., "Service Orientation: Winning strategies and best practices," Cambridge University Press, 2006

[6] McGovern, J., Tyagi, S., Stevens, M., Matthew, S., "Java Web Services Architecture," Morgan Kaufmann Publishers, 2003

[7] Hurwitz, J., Bloor, R., Baroudi, C., Kaufman, M., "Service Oriented Architecture for dummies," John Wiley and Sons, 2007

[8] Bieberstein, N., Bose, S., Flammante, M., Jones, K., Shah, R., "Service-Oriented Architecture Compass-Business Value, Planning, and Enterprise Roadmap" IBM Press, 2006

[9] Khoshafian, S., "Service Oriented Enterprise," Auerbach publications, 2007

[10] Hoof, J.V., "SOA and EDA: using events to bridge decoupled service boundaries," SOA Magazine Issue IV, February 2007.

[11] Michelson, B.M., "Event-Driven Architecture Overview" Patricia Seybold Group, 2006

[12] Schulte, R.W., "Tutorial for EDA and how it relates to SOA" Gartner, 2008

[13] Ranadivé, V., "The power of now. Computing," McGraw-Hill, 1999