Autonomous Event Driven Architectures upon Service Oriented Architectures

T. S. L. Fernando[1], W. W. M. E. P. Gunetileke[1], H. Halgaswatta[1], S. Weerawarana[2], P. Fremantle[3], I. Perera[1]
[1]Department of Computer Science & Engineering, University of Moratuwa, Sri Lanka.
[2]WSO2 Inc.,Sri Lanka.
[3]WSO2 Inc., United Kingdom.

Abstract— **Event Driven Architectures (EDA) and Service Oriented Architectures (SOA) are two technologies that have both gained much popularity among IT infrastructure in worldwide businesses. Although they do have their differences, businesses are discovering that the two actually complement each other, and SOA can be effectively utilized as a platform in implementing an EDA. This paper explores this concept and also highlights Project SCI-Flex; which is a venture towards integration of SOA and Complex Event Processing. Here we look at how the SCI-Flex middleware system may be used in implementing Autonomous and fully functional EDAs.**

Index Terms— **CEP, EDA, SOA**

I. Introduction

EVENT Driven Architectures (EDA) are basically systems that disseminate relevant or notable events or occurrences within a business and its environment to all interested parties and allow evaluation of this event followed by necessary action. The concept is certainly nothing new, and even the presence of EDA in IT systems is now decades old.

In recent years, with the needs for an IT infrastructure with distributed computing capability due to global business model, as well as high flexibility, scalability and interoperability among varied platforms, the adaptation of Service Oriented Architectures (SOA) has become increasingly popular. In fact at this point, most major businesses and ventures would likely either be involved with SOA on some aspect in their business infrastructure or are planning towards the implementation.

Along with these developments, an observation coming into light is that EDA and SOA highly complement each other. It has been noted that SOA offers many opportunities as an implementation platform upon which EDAs may be constructed, as well as the inherent features of SOA would well facilitate EDA implementation. The distinction of these technologies therefore is increasingly becoming blurred.

Speaking of the developments we have made in our project: SCI-Flex is a into the seamless merging of SOA and Complex Event Processing. With the CEP style inherently being a vital part of high end EDAs, we have therefore recognized the opportunity for utilizing the SCI-Flex platform towards the aspect of EDA implementation within an SOA infrastructure.

Taking this one step further, we have seen that it be possible to implement this system with a unique level of Autonomy within the EDA. Autonomy naturally being a major goal in many aspects of any business, and more so in businesses employing EDA models, there will be a large pool of opportunity for this model in the real world [1].

–Section Two: 'Background' will describe some of the major concepts used in this paper. Readers already familiar with the realms of SOA, CEP and EDA systems will find this to be basic knowledge on the subject matter and thus may commence from the next section.

–Section Three: 'EDA on SOA' will basically look at how we do SOA on EDA and will look at the combining and co-existence of these two concepts.

–Section Four: 'Autonomy in EDA with SOA' will bring in the third important topic in this context into play and will aim to describe the importance of Autonomy in EDA systems and how Autonomy can take place in the EDA with SOA environment we have defined.

–Section Five: 'Implementation' will explain how our system aims to achieve the goals of autonomy in this context.

–Section Six: 'Relevance To Existing Systems' will be a look at how the concepts in this paper will play out in the real world.

–Section Seven: 'Future Work' will highlight possible improvements and new developments that could take place in our project.

–Section Eight: 'Related Work', will be a similar section where we will explore systems and implementations that are similar to the objectives we put forth here.

II. BACKGROUND

This Section will aim to briefly introduce the important topics and concepts in this research paper, thereby laying the foundation to build upon for the content to follow.

A. Service Oriented Architecture (SOA)

A service-oriented architecture is essentially a collection of services. These services will then communicate with each other. Here, this communication can involve either simple data passing or it could involve two or more services coordinating some activity. Some means of connecting services to each other is needed. Service-oriented architectures are not a new thing. The first service-oriented architecture for many people in the past was with the use Distributed Component Object Model (DCOM) or Object Request Brokers (ORBs) based on the Common Object Request Broker Architecture (CORBA) specification.

B. Services

If a service-oriented architecture is to be effective, we need a clear understanding of the term service. A service is a function that is well-defined, self-contained, and does not depend on the context or state of other services. The following figure illustrates a basic service-oriented architecture. It shows a service consumer at the right sending a service request message to a service provider at the left. The service provider returns a response message to the service consumer. The request and subsequent response connections are defined in some way that is understandable to both the service consumer and service provider.

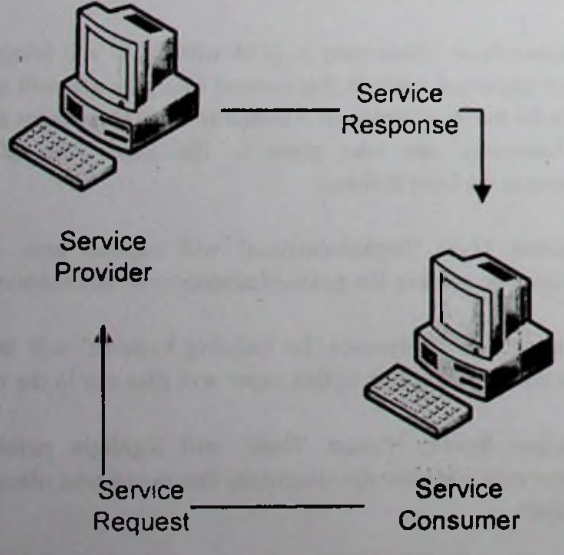

Fig. 1. Service Oriented Architecture

C. Event Driven Architecture (EDA)

Event-driven Architecture (EDA) is a style of application architecture centered on asynchronous "push"-based communication. EDA is the architecture of choice for implementing "straight-through" multistage business processes that deliver goods, services and information with minimum delay. Processes designed using EDA are also easier to modify than traditional applications [2].

D. Complex Event Processing (CEP)

Complex-Event Processing (CEP), a sophisticated form of EDA, is used to extract the information value from multiple events. CEP systems find patterns in event data to detect opportunities and threats. Timely alerts are then pushed to the appropriate recipients, often using Business Activity Monitoring (BAM) dashboards. The result is faster and better operational decisions and more timely responses.

E. Autonomy

Autonomy is basically the capability of a certain system to take independent intelligent action in response to certain scenarios without the need for constant human intervention in order to make the necessary decisions.

Taking this to the Web Service arena, Autonomous Web Services Web Services represent the most recent as well as the most revolutionary technology developed for the World Wide Web. Web Services are transforming the web from a collection of static pages to a web of dynamic service providers that automatically discover information that we seek, negotiate on our behalf for goods we intend to purchase, gather information from different sources, and fuse it into coherent forms. Today, Web Services are discovered and invoked manually by human users, limiting the ability to take advantage of opportunities that may exist.

Our efforts are aimed at overcoming these limitations, allowing Web Services to discover and interact with each other autonomously, thus reducing the need for human mediation. Not only this mere autonomy of services, but to allow all this to happen within an EDA empowered by SOA.

III. EDA ON SOA

At this point, having understood the basic principles and features of both SOA and EDA, it is not difficult to imagine a coexistence and interoperation between the two technologies. Let us consider the basic concept of an Event itself in EDA. For the dissemination of Events which would trigger any suitable response in the given system, a number of methodologies may be utilized such as certain messaging middleware, a publish/subscribe model or any such means of establishing event notification. This core feature itself can easily be extended onto the Web Service arena where we will encounter methodologies of WS-Eventing and WS-Notification. Here itself we see the potential level of interoperation between SOA and EDA [3].

Considering the broad environment of SOA as well as the recent developments in the field, it is technically correct in a way to say that EDA can actually be a subset of SOA. We can start off on this pattern of thinking from the point discussed in the previous paragraph: that the event notification models which lie at the core of an EDA are naturally a part of messaging methodologies of SOA such as SOAP. Although, such methodologies traditionally may not provide the full set of features to implement a full Event-Driven environment, developments in related standards such as WS-Notification, WS-Topics and WS-BusinessActivity; enable the implementation of a fully fledged event driven platform within the realm of SOA based on Web Services.

IV. Autonomy in EDA with SOA

Autonomy as mentioned earlier is a key objective in a large number of business processes in virtually every industry. Specifically, IT-aided autonomy has for many years been an enabler of autonomy in this realm. Thinking about EDAs in general, it is obvious that a greater degree of autonomy concerning event notification and response would be a major goal.

We can see that this added degree of autonomy can be a key possibility in the merging of EDA on SOA. One distinction in general between SOA and EDA is the level of coupling. EDA" s can generally be implemented at a far greater level of decoupling compared to an SOA, where some level of loose coupling such as service agreements may be utilized. However, considering the practical business needs for today' s event driven systems, a full decoupling may cause inadequacies when it comes to communication among applications. With a generic decoupled methodology, EDAs will have problems in creating application-to-application autonomy in systems; and thus from a business perspective, it is important that they support a certain level of coupling in order to enable greater autonomy. This again makes an SOA a prime candidate toward the goal of establishment of an EDA.

We must also take into account the trend that businesses are today evolving into a more flexible and scalable frame of operation which include a multitude of service providers that act autonomously and freely cater to service consumers either internal or external to the business itself. This trend is supported by the shift in paradigm towards becoming an "on-demand" business whereupon the ability to quickly respond to Events from the environment is vital. This would obviously require a high degree of autonomy throughout the system. All of this we can well see, points towards an Autonomous EDA implementation upon an SOA platform which would facilitate all these requirements [4]. The key point to note here is that for all practical purposes, Services in an SOA are indeed autonomous. Services naturally maintain a private inner structure and aim to be self contained in general. This fits in perfectly with the EDA model we are discussing. Enjoying the benefits of Autonomy upon an SOA, EDAs can further enhance their Autonomy by caching the service states.

V. Implementation

Project SCI-Flex includes the capability of running CEP queries on a distributed system based on a scalable SOA infrastructure. Our system supports autonomy in a partial sense, meaning that it initially requires some amount of configuration detail which can then be used for the system to evolve on its own, based on various stimuli generated in the outside environment.

At the start, we will have a certain pre-defined rules (which in fact are CEP queries) that will make the system up and running, based on these initial rules the system will interact with external systems and respond to both internal and external events. It is intelligent enough to understand what actions it needs to perform for each interesting event it identifies, and it is also capable of understanding what events are important and what are not, by means of the powerful CEP capabilities the system possesses. The autonomous behavior is supported by the system being able to reconfigure itself based on the events it identifies. The reconfiguration will invariably make the system react in a way that is very different from what it has been thus far and therefore, the system is capable to adapt to rapid changes in the environment without the need of any human or other intervention.

As it seems the system is capable of responding as well as changing accordingly to stimuli from its external environment. Such a stimulus can be a change of a system property or even a failure in a connected node, or can even be some random error to which the system must adopt to. Therefore, SCI-Flex can also to a greater extent be used as a driver of fault tolerance in a typical SOA infrastructure. However, the system itself is not fault tolerant, and can in fact be a central point of failure should it not be properly functioning. We have overcome these limitations by making it possible for SCI-Flex to collaborate with similar systems, and thereby create a cluster of nodes that can act as a fault tolerant system [10]. For this to be realistic the various nodes of the cluster must in fact be geographically dispersed systems. We also provide the capability to add rules that can perform routine checks to avoid a potential occurrence of the split brain syndrome [11].

VI. Relevance To Existing Systems

Here we will look at applications or systems in the real world which can make use of the Autonomous EDA in SOA concept to either boost their capabilities or to expand their range of application as it stands of now.

A. Business Process Execution Language (BPEL) Applications

Taking a more overall point of view of SOA and EDA

implementations, a large range of applications can be found in the realm of Business Process Management and Business Intelligence. Business events which increasingly need to act upon changing environments or external factors are a key driver in service integration in today's distributed computing implementations. Autonomous EDA obviously provides a key leverage to this arena [5].

B. Trading Platforms

Looking at more specific applications, the realm of creating software and systems for usage as trading platforms in local or global markets is one to which this concept has a huge relevance.

Current trading platforms work in an environment of massive, constantly changing event streams with the need to for quick responses to predefined events. The application of autonomous EDA technology to the distributed environment of trading platforms would enable the creation of even more intelligent trade management systems. With the application of Complex Event Processing power, trading platforms can be made more efficient and more complex in identifying and acting upon complicated event patterns or environment changes.

C. Network Traffic Analysis

Network monitoring applications constantly need to monitor large streams of events, deal with rapid changes in this event stream and give a speedy response suitable for identified anomalies or pre-defined patters. With the advent of massive worldwide networking that is becoming pretty much a norm at current times, this is a task which is becoming increasingly complex and difficult.

These various Network Traffic Analysis tools or Network Monitoring applications are all existing systems upon which the concepts and technologies of autonomous EDAs upon SOA environments can really make an impact. With the application of these concepts, and (especially as brought forward in the SCI-Flex framework) Complex Event Processing capability, it will be able to vastly increase the processing capabilities and general intelligence of network traffic analysis. Engaging this technology will create network analysis tools which are more efficient, more responsive and can be tailored for much more complicated activity.

D. Transportation Service Management

It is also noteworthy to make a quick mention of the potential application of this technology towards the area of transportation service management. With the use of autonomous EDA's at the controlling hubs or nodes of massively distributed transport services, the entire process of managing service requirements can be made more efficient. As a more specific example, it will be able to make systems more responsive to customer service requests by quickly processing locally available transport service provisions and autonomously allocating the necessary resources.

As a subtopic here, we can consider existing Air Traffic Control systems which can be further enhanced by the applications of autonomous EDA over the SOA environment. With the massively increasing amounts of air traffic, the managing systems may become slower in response to individual requests and the possibility of traffic congestions may also be on the rise [6]. Multi-agent systems [7] are being considered for deployment as a solution for this increasing load, and this type of agent based monitoring system can be easily created with the technology presented in this paper.

VII. Future Work

As of now SCI-Flex requires tedious configuration, and also a great deal of knowledge on how to setup an autonomous EDA, in a SOA infrastructure. Clearly, we also do need to come up with some generic interface-based approach for configuration in addition to the monitoring capabilities that we provide. Also, it is interesting should someone come up with an implementation that helps a SCI-Flex instance automatically configure itself as it plugs into an existing SOA infrastructure require 0% configuration at startup.

We also would like to embed intelligence and learning capabilities to this system so that it also could learn out of the various information it gathers out of events instead of a simple stimulus-response operation that works out of the box. Also, SCI-Flex is yet to be deployed in a very large scale system that handles massive amounts of information which the system supports in theory but has yet to be exploited. We are interested in improving the system's performance, especially for very high levels of concurrency.

VIII. Related Work

A. EDA for Enterprise Reconfigurability and Optimization

Looking at autonomy in the realm of EDA, one area related to the current topic is the reconfiguring and optimization of business processes in various levels of an enterprise [8]. This will be a combination of System Tasks such as Optimization, Analysis and Concurrent Estimation; Distributed computing related tasks such as Sensor Network Adaptability and Node Localization as well as Event Detection tasks such as Information Acquisition and Anomaly Detection. In relation to the concepts presented in this paper, we may see a merging of these concepts towards more efficient working of this application over wider distributed networks. Also, greater possibilities exist in the application of Complex Event Processing to the whole realms as well in order to create a greater responsiveness and better intelligence.

B. Multi-Agent And Event Driven Based Dynamic Collision Avoidance

We mentioned in section 6 of this paper that multi-agent

systems were being applied as a future onset for Air Traffic Control Systems. Taking this concept a bit deeper, we see that there is much work and research being put into this area especially towards the developed of more efficient and accurate collision avoidance systems [9]. This work is progressing in many areas ranging from collision avoidance in automated robotics all the way to the aerospace industry as mentioned here. The concepts presented in this paper definitely have a bearing on this realm of work and without a doubt with both the requirements for greater distributed computing and greater event processing capability on the rise, we will see autonomous Event driven Systems working upon Service Oriented Architecture environments in this area.

REFERENCES

[1] Jason Bloomberg, "SOA + EDA = FUD?", June 2004

[2] Brenda M. Michelson, "Event-Driven SOA Is Just Part of the EDA Story", 2006 Patricia Seybold Group and Elemental Links, Inc, http://dx.doi.org/10.1571/bda2-2-06cc

[3] Daniel Rubio, "How EDA and CEP fit in with SOA", June 2007

[4] Jack van Hoof, "SOA and EDA: Using events to bridge decoupled service boundaries", June 2007

[5] Paul Vincent, "CEP vs WSDL + SCA + BPEL", August 2008

[6] Kagan T. - Oregon State University, Agogino A. UCSC, NASA Ames Research Center, "Distributed Agent-Based Air Traffic Flow Management" , AAMAS'07 May 14–18 2007, Honolulu, Hawai'i, USA.

[7] Menon P.K., Sweriduk B. "Optimal Strategies for Free-Flight Air Traffic Conflict Resolution" , Journal of Guidance, Control, and Dynamics, 1999 http://www.optisyn.com/papers/1997/atc_97.pdf

[8] Christos G. Cassandras, Boston University, "Event-Driven Sensing For Enterprise Reconfigurability And Optimization", 2006

[9] Becker, Raphen; Lesser, Victor; Zilberstein, Shlomo, " Decentralized Markov Decision Processes with Event-Driven Interactions", Mutli-Agent Systems Lab (http://mas.cs.umass.edu) , 2004

[10] Nan Du; Bin Wu; Bai Wang, "Concept Forest: A New Ontology-assisted Text Document Similarity Measurement Method", Web Intelligence, IEEE/WIC/ACM International Conference, Nov. 2-5, 2007.

[11] Shum K.H., "Fault tolerant cluster computing through replication", Proceedings of the 1997 International Conference on Parallel and Distributed Systems, 1997.