The Relationship between Service Oriented Architecture and Enterprise Architecture

Christopher Kistasamy
Finance, Risk, and Treasury Directorate
ABSA Bank of South Africa
Johannesburg, South Africa
e-mail: christopher.kistasamy@absa.co.za

Alta Van Der Merwe
Meraka Institute
CSIR
Pretoria, South Africa
e-mail: alta@meraka.org.za

Andre De La Harpe
Faculty of Informatics and Design
Cape Peninsula University of Technology
Cape Town, South Africa
e-mail: andre.delaharpe@i2icentral.com

Abstract - The adoption of Enterprise Architecture (EA) concepts within organizations is causing an interest in the methodologies and supporting technologies available. Service Oriented Architecture (SOA) supports EA in many facets. However, there is much dissolution with regard to the relationship between EA and SOA within organizations. There are also potential problems that may arise if this relationship between SOA and EA is not agreed to at the outset of implementing an EA. The purpose of this paper is to discuss the growing role and importance of understanding the relationship between SOA and EA, and in doing so to demystify some of the expectations of the role that SOA plays in EA.

Keywords - Enterprise architecture; Service Oriented Architecture; Relationship between Enterprise architecture and Service Oriented Architecture.

I. INTRODUCTION

Pieter Idoine, MD of Oracle New Zealand said that: “The role of the Enterprise Architect is to intervene for the benefit of mankind”[1]. Although we can find a myriad of research around EA, this statement is one that defines the essence of EA and provides a very thought provoking question as the understanding of EA in today’s world. EA defines how business and Information systems alignment should be achieved [2]. It is where business capability (financial and market goals) and technology capability (products, vendors, and functionality) are tied together with organisational capability (people or process) to drive an ongoing strategy or desired outcome [1]. SOA is recognised as a methodology optimised in applications architecture or Service architecture [3], with a view to deliver one of the domains within EA namely the applications architecture. The understanding of SOA has however evolved over the years, representing itself in other domains of architecture such as the business architecture where SOA focus on analysing the business to identify business areas and business processes by defining services that represent them [4]. SOA and EA both aim to address the issue of alignment between business and IT, however in many instances this is both difficult and costly to accomplish without understanding the relationship between them. In most organisations, alignment is also only dealt with at a strategy level, and even research into the role of the Chief Information Officer talks about aligning the IT strategy with the business strategy. It is therefore assumed that if we align at a strategy level then we will also have alignment between business and IT at a technology level; however this is most often not the case. In order to truly embrace the real value of EA, it is imperative that alignment between business and IT is achieved at both the strategy and technology levels – EA provides alignment at the strategy level and SOA provides alignment at the technology level. SOA is part of EA [5]. Many SOA and EA practitioners as well as organisations today struggle with the notion of practically relating SOA and EA, leading to a lot of confusion for businesses and inevitably unnecessary spend on projects. Both SOA and EA are distinct disciplines that share a number of common goals, the most noticeable being the fact that they both promise improved interoperability as well as better alignment of business strategy and Information Technology solutions [6]. In order to fully understand the relationship between SOA and EA, it is imperative that the understandings of both these disciplines are based on the same end goal and for the purpose of this paper the sections that will follow are based on the premise that both disciplines are business tools where EA focuses on enabling the strategy of the business and SOA focuses on providing the business capabilities through technology solutions that enable EA with great emphasis on reusability. The purpose of this paper is to provide an overview of the discipline of EA and SOA and to highlight the relationship between them. In the sections that follow,
the paper focuses on the relationship between SOA and EA, taking into account the similarities and differences as well as the role of SOA in EA projects and role of EA in SOA projects. In the last section the conclusion focuses on some pitfalls that may arise if the relationship between SOA and EA is not addressed.

II. OVERVIEW OF EA

Alignment between business and IT within an organization is a fundamental challenge facing all organizations today [7]. Over the years this has been the most fundamental challenge for most organisations, the question of how to justify the IT budget especially when there seems to be very little evidence stating that this supporting function is adding sufficient value to the bottom line of the organization especially in non-IT businesses. EA is about understanding all of the different elements that go to make up the enterprise and how those elements inter-relate. EA embodies a set of principles, rules, standards and guidelines, expressing and visualising the vision, culture & behaviour of an organisation while implementing certain concepts that serves as prescription for the design and construction of a certain object type. It contains a combination of style, engineering and construction principles, guaranteeing the uniformity and quality of the resulting object [8]. A successful EA should demonstrate the following characteristics within an organisation [9]:

- adherence to the corporate policies
- adherence to the corporate processes
- comply with laws, rules and regulations
- conform to auditable controls

Table 1 defines simplified EA and provides a high level understanding of the different aspects that are involved [10].

III. EA FRAMEWORKS

EA Frameworks, made popular first by John Zachman, and since by many different EA framework creators, tool vendors and practitioners, appear to be a very polarizing phenomenon. There are many devotees to a particular framework; others who are certain you must have one, but not sure which one to pick or how to use one they have chosen; others who could take them or leave them; with still a few who would tell you they are a big waste of time [11]. In order to define requirements for the development of EA descriptions we have to identify elements of a method, which are essential to differentiate between methods and other approaches [12]. Each framework or method must contain the following five constitutive elements [12]:

- Meta Model
- Procedure Model
- Technique / Modelling
- Role
- Specification Document

Table 2 outlines the major frameworks and evaluates them in terms of the five constitutive elements that Leist and Zellner speak about. It is also interesting to see that at the time of evaluation none of the frameworks mentioned were able to incorporate all of the elements into their model. Many organisations have adopted a combination of frameworks in terms of their EA practices, mainly due to the fact that no one framework has all the elements that every organisation might need. Table 3 highlights the benefits as well as the shortcomings of EA frameworks based on the work done by EA Directions [11]. As reflected in Table 3 EA frameworks have both advantages as well as many disadvantages and it is imperative that organisation to not fall into the trap of allowing a framework to drive the EA practices. The frameworks are there to guide the implementation and to assist with creating the various views of the organisation, but it can only be useful if it is driven by the goals and objectives of the organisational implementation.

IV. OVERVIEW OF SOA

Service Architecture, Service Orientated Architecture (SOA), and many other architectural buzzwords abound in technology at the moment. Although the standard view is that Services are a technology solution to add agility, past experience teaches us that technology solutions rarely deliver agility except when they are focused on the business visions [13]. As with any fairly new concept in the world of IT, there are numerous definitions, however this paper makes use of the definition below specifically to outline the fact that as the author states, SOA is a business tool that allows business to architect their processes independent of each other, thus driving the other architectural domains to follow the same logic. The definition provided is as follows: “Service Orientated Architecture (SOA) is 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” [14]. In order to fully understand the concept of SOA it is best to break up the concept as per the terms that make it up, namely: Service and Architecture, the orientation bit really speaks about designing architecture as a service. The objective of a service is to represent what the business does and place a boundary which all parties, but predominately the business can agree on, it is this representation of the business that the creation of a Service Architecture must be focused, technology is very much a secondary element [13]. In essence, the above statement forms the foundation of what SOA is really about – it is the aligning of the technology with the business processes and ensuring that these elements are constantly monitored so that the technology is always addressing the business functions. The second part of the concept of SOA is the architecture which is defined as: “architecture is the structure of the system, comprised of components or building blocks, the external visible properties of those components, and the relationships among them” [15]. If one where to look at the definitions of both terms that make up SOA, then one can deduce the following: SOA is the process of representing what the business does in the context of the structure of the systems, which are comprised of components or building blocks, the external visible properties of those components and the relationships among them. The following properties have been identified and must be prevalent for an architecture to be referred to as a SOA [16]:

- architecture is the structure of the system, comprised of components or building blocks, the external visible properties of those components, and the relationships among them
- architecture is the structure of the system, comprised of components or building blocks, the external visible properties of those components and the relationships among them
Discoverable and Dynamic – Services are discovered using a directory. Services are bound at runtime, rather than compile time.

Loosely Coupled – SOAs are composed of multiple services connected in such a way as to be resilient in the face of network failures and latency. This loose coupling gives SOAs a distinctly different architecture than programs that are distributed, but still connected synchronously and in ways that make the overall system brittle.

Locationally Transparent – SOAs are constructed in such a way that the overall system is unaware of, or at least ambivalent to the location of various services.

Diversely Owned - SOAs may be composed of services which are owned and operated by outside organizations. Diverse ownership implies that the published service interface will be treated as a black box from the standpoint of the programmers since they cannot penetrate the interface and modify code and behaviour behind it.

Interoperable – Standards ensure that services from differing organizations can use each other’s services.

Composable – Applications in SOAs are created by composing pre-existing, well-tested services from multiple providers.

Network-addressable – Networks are central to the idea of services that are discoverable and interoperable. This allows applications to be composed that run on different machines.

Self-healing – When applications are created by composing dynamically discovered components that are owned by multiple organizations, the ability of the system to rediscover and bind to working services when services fail is critical.

Although these are quite technical characteristics, they key here are the principles around these concepts, whether we address SOA from a technology or from a business perspective these principles must prevail. Figure 1 provides a high level overview of the SOA components and the relationships between them. This is at an execution level however these elements are the real essence of SOA, but can only be activated and implemented effectively once we have a defined business problem to work off. Figure 1 aims to highlight the elements involved in designing and implementing a single services, the processes involved between the service consumer as well as the service, and finally the interfaces that exist between the service consumer and the service, including the relationships between those interfaces.

V. THE BUSINESS REASON FOR SOA

Eero Saarinen, a Finnish American Architect famous for varying his architecting style based on the demand of the project, said:” Always design a thing by considering it in its next larger context –

- A chair in a room
- A room in a house
- A house in an environment
- An environment in a city plan. ”

The process of SOA is based very much on the above design principle where each business process or function must be considered in its larger context so as to ensure that at the end of it all the design provides an organisation with a vehicle to operate efficiently and the technology is able to change as and when the business changes whether it be as a result of internal or external factors. A service architecture broadly follows a four step process, or more accurately delivers three stages of the process and provides the direction for the fourth [13]:

What: Defining the scope of services, this about determine what the services actually are.

Who: Who are the external actors that drive the services or with which the services interact.

Why: Identifying why one service talks to another, and why external actors interact with the services

How: The detail about the processes that co-ordinate the services and also the detail on how a service itself will be implemented.

Figure 2 provides a view of the interactions between IT and business and where the possible links are found and this is one of the perspectives that SOA needs to address which is to identify all of the links between business and IT from a people, process and technology context. Figure 2 could become much more complex if one were to break up the larger components into smaller ones. The key aspect in Figure 2 is that it deals with the skills that are in use and not the domain under which they find themselves. The aim of the Figure 2 is to depict a key element to the success of SOA and that is to provide an answer to the “What”? SOA projects that are technology focussed tend to concentrate on the how rather than determining the reason for embarking on an initiative. This approach has proven costly in many instances where SOA projects have not added any value to organisations simply because the business reasons behind an implementation are not taken into account.

There are a number of technical debates and discussions especially around the topic of SOA and the intricacies involved in the provisioning of services and the integration of legacy systems into an organisation’s SOA. In terms of this paper the aim here is to position SOA in the context of the business and to understand the business case for a SOA implementation. It is my view that SOA should never be considered unless there is a business need for it that can be substantiated by a relevant business case. Gartner advises that Service-Oriented development of applications is estimated to reduce total IT expenses over the long term by as much as 20% compared to traditional development methods and the savings becomes exponential over time as library of business services expands and greater degree of reuse is achieved. However, looking only at the cost savings misses opportunities for the dramatic business process improvement SOA can deliver [17]. The promise of real value has always been a contentious subject when it comes to SOA; however there are many organisations that have seen enormous value when using an SOA approach to delivering projects. SOA gives rise to both business and technology benefits and Roch [17], elaborates on these in Table 4. The differences between the business and technology benefits of SOA can be clearly seen in Table 4; however it is very important to understand that even though SOA does provide technology benefits – it should never be implemented.
VI. RELATIONSHIP BETWEEN SOA AND EA

EA and SOA are integral strategic disciplines that provide organisations with a tangible means of aligning business and IT, however due to the misunderstanding of the relationship between the two disciplines; very few organisations have seen the combined benefit. Both SOA and EA practitioners have also not made things simpler by adding the extra element of rivalry between camps so as to suggest that one discipline is more important than the other or that one camp should lead the other or that both can do each other’s jobs. The rivalry, although misplaced, has created very little synergy between EA and SOA efforts, in the end causing the clients to suffer the consequences either of having projects delivered late or having to come in way over project budget. Some traditional enterprise architects have not done a stellar job in understanding the opportunities within SOA, generally speaking, and the SOA practitioners have not figured out how SOA meshes with existing EA standards, notations, and practices [5]. EA supplies answers that encompass organization-wide processes, how they are aligned with corporate strategies and how processes are linked to performance measures. SOA offers customer-responsive IT solutions and inter-functional coordination of information flows and a robust platform upon which the business can apply their applications [18]. EA and SOA clearly can co-exist based on this definition, and one can also see that both are equally significant in ensuring the technology solutions are supportive of the business processes, in fact one can argue that both have a clear role to play thus alleviating any confusion, however this is much easier said than done due to the similarities between the two concepts. Figure 3 provides a high level view of the different forces that are at play in the current industry that impact the way in which enterprises are architected. It is imperative that the business drivers are clear and understood when embarking on an EA implementation, so as to ensure that that architecture is not built for every particular problem. For example, to software developers SO can be applied to various disciplines; however it is simply a design style and depending on the implementer there will be a number of different ways to develop a solution to any particular problem. For example, to software developers SO will relate to how programming constructs as to the definition of how to develop and publish a service, however to business people, SO would relate to business strategies that leverage reuse such as the implementation of a common tax standard across a business. Table 5 maps out the elements of the SOA solutions stack to their corresponding domains within the EA Framework. This is key to the understanding of the relationship between EA and SOA as it contextualises the terminology as well as the practices that are involved in both EA and SOA, however the key question then becomes can one exist without the other. SOA is not concerned with the development of business architecture. Instead, it uses the outcome of business processes and other business architecture artefacts, such as Component Business Modelling (CBM), as input to identify business services. In contrast, EA is concerned with the development of business architecture, including business processes [20]. According to [20], EA and SOA are complementary with each dealing differently with the domains of EA. Table 6 highlights both the similarities as well as the differences between SOA and EA. Advances in integration technology, primarily intelligent and flexible middleware and web services, are providing new ways for designing more agile, more responsive EA that can provide the kind of value businesses seek. With new architecture, IT can build new business capabilities faster, cheaper and in a vocabulary the business understands [21]. There are natural synergies that exist between SOA and EA, and in today’s world most EA projects utilise some form of SOA as it provides a method with which to quickly align technology or applications with the business architecture. Table 7 provides a project view of how these two concepts should complement each other; however the same principles can also be used programmes as well as strategies in terms of aligning SOA and EA.

VII. DISCUSSION

This paper forms part of a much larger research initiative with the objective of investigating the role that SOA plays as an enabler for EA. The premise that is used as a basis for the research is that EA is used to align business and IT from a strategic perspective and SOA is used to align business and IT from a technology perspective. Traditional EA focuses on the crafting of a plan and not the implementation of a solution whereas SOA focuses on the implementation once the planning has been done [22]. Even though both concepts have been in industry for quite some time, there still seems to be confusion as to roles of each and how they relate to each other or even whether they relate to each other at all. Linthicum [22] highlighted the following issues as some of the major contributors to the confusion between the role of SOA and the role of EA:

- Traditional EA focuses on the creation of a strategy rather than implementing a strategy. SOA addresses the implementation after the planning has occurred.
- Traditional EA is not funded for implementation; the focus is on the creation of the strategy, the plan or the means with which to move an organization forward strategically. Moreover, many enterprise architects just seem to serve as the resident guru and weld no real political or budgetary power.
- Traditional EA is leveraging well-defined processes, approaches, and methodologies that in some instances are difficult to map into SOA.

Both SOA and EA are distinct disciplines that share a number of common goals, the most noticeable being the fact that they both promise improved interoperability as well as better alignment of business strategy and Information Technology (IT) solutions [6]. Although
both EA and SOA focus on aligning business and IT, the alignment is achieved through different mechanisms each of equal importance in achieving the vision of business and IT alignment.

VIII. CONCLUSION

The objectives of EA and SOA are quite similar. However, EA is a framework that covers all dimensions of IT architecture for the enterprise, while SOA provides an architectural strategy that uses the concept of “Service” as the underlining business-IT alignment entity[21]. Both concepts aim to provide a guideline as well as a process of ensuring that an organisations business practices are fully supported by its investment in IT. There are however potential problems that may arise if this relationship between SOA and EA is not agreed to at the outset of implementing an EA, most notably [20]:

- Potential overlap between the responsibilities of the SOA lead and the enterprise architect. This overlap in responsibilities may cause confusion and friction between the two leads that ultimately might impede the success of both SOA and EA.
- Competition between SOA and EA for the same business resources. In most enterprises, the time and availability of business subject matter experts are scarce resources. As such, asking them to participate in duplicate and similar activities and governance organizations for both SOA and EA can cause lack of participation and perception of inefficiencies. This can lead to less contribution by these experts to the activities of one or both.
- Potential for making contradicting architectural decisions that affect the whole enterprise. With both efforts for SOA and EA progressing in isolation, it's likely that some of the decisions made by one or the other could cause further confusion among those who are relying on the outcome to guide their decisions.

In my experience with different industries both in the public and private sector, it is very clear that IT professionals have completely missed the opportunity that SOA provided when it became mainstream, mainly due to the fact that IT professionals viewed the discipline purely from a technical perspective and never really considered the potential business benefits from a business perspective that SOA could provide. If we remove the technical jargon, and we look at SOA from a purely conceptual perspective, it has the potential to provide solutions to the age old question of what do customers want. Customers in any sector or industry want SERVICE!!! – SOA provides a framework that is driven by the creation of services. Furthermore, in this 21st century that we are living in, it is imperative that organizations understand that IT is no longer an enabler, but has now become a differentiator with the potential to provide competitive advantage to organizations that embrace this fact. A case in point is if we look at Internet banking, in the banking sector. The Internet as a platform created a case for where technology would drive business, and in the banking sector the Internet allowed for the creation of an entirely new channel for banking. Customers required convenience banking, the internet provided the platform for this – and the banks created the Internet Banking service. This is an example of what the true values of SOA and EA can bring when used in the correct contexts. I have not delved into the detail of this scenario but I am sure one can see the potential of this type of framework of thinking. In the event that organisations begin to embrace this conceptual framework, there is definitely great potential for further research into this area, especially with regards to creating a new platform for true SOA, not in the technical sense but in a way enables business to provide value to their customers through services. This paper has provided a high level discussion of the relationship between SOA and EA, clearly indicating the close relationship between the two concepts, in fact highlighting the point that these two concepts should not be dealt with in isolation to each other especially when dealing with enterprise wide projects.

IX. FIGURES AND TABLES

<table>
<thead>
<tr>
<th>WHAT</th>
<th>HOW</th>
<th>WHERE</th>
<th>WHO</th>
<th>WHEN</th>
<th>WHY</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENTERPRISE MODEL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Owner</td>
<td>- Resource model</td>
<td>- Business Process models</td>
<td>Business logic System</td>
<td>- IT goals aligned with strategy and goals</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Information model</td>
<td>- Business rules - IT%</td>
<td>Organization models</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSTEM MODEL Designer</td>
<td>Logical Data model</td>
<td>Application Architecture</td>
<td>Distributed System Architecture</td>
<td>Human Interface Architecture</td>
<td>Processing Structure</td>
</tr>
<tr>
<td>TECHNOLOGY MODEL builder</td>
<td>Physical Data model</td>
<td>System Design</td>
<td>Technology Architecture</td>
<td>Presentation Architecture</td>
<td>Control Structure</td>
</tr>
</tbody>
</table>

Table 1: Simplified EA [10]
### Table 2: Evaluation of Current EA Frameworks [12]

<table>
<thead>
<tr>
<th>EA Framework</th>
<th>ARIS</th>
<th>C4ISR/DoDAF</th>
<th>FEAF</th>
<th>MDA</th>
<th>TEAF</th>
<th>TOGAF</th>
<th>Zachman</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specification document</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>○</td>
<td>●</td>
</tr>
<tr>
<td>Meta model</td>
<td>●</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>●</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Role</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>●</td>
<td>○</td>
<td>●</td>
<td>○</td>
</tr>
<tr>
<td>Technique</td>
<td>●</td>
<td>●</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Procedure model</td>
<td>○</td>
<td>●</td>
<td>●</td>
<td>○</td>
<td>●</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>

**Legend:**
- ● Fully accomplished
- ○ Partly accomplished
- ○ Not accomplished

Table 2: Evaluation of Current EA Frameworks [12]

### Table 3: EA Framework Benefits and Shortfalls [12]

<table>
<thead>
<tr>
<th>EA Framework Benefits</th>
<th>EA Framework Shortfalls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simplify the complexity of the entities they represent.</td>
<td>They are too process centric in their make-up – this can become very time-consuming.</td>
</tr>
<tr>
<td>Presents a simplified version of that complexity for communication, analysis, and deliverables.</td>
<td>Assume that they have the ability to think for an organisation.</td>
</tr>
<tr>
<td>Assists to organise the huge amount of complex entities and relationships that make up an enterprise.</td>
<td>Can be very rigid on what can be modelled without taking into account the organisations needs.</td>
</tr>
<tr>
<td>Highlights all the areas to consider for the scope of an EA.</td>
<td>Always seems to prefer the big bang approach as opposed to smaller chunks at a time.</td>
</tr>
<tr>
<td>Provides customisation opportunities to suite organisations needs.</td>
<td>Does not foster creativity due to the rigidity.</td>
</tr>
</tbody>
</table>

Table 3: EA Framework Benefits and Shortfalls [12]

### Table 4: SOA Technology and Business Benefits

<table>
<thead>
<tr>
<th>SOA TECHNOLOGY BENEFITS</th>
<th>SOA BUSINESS BENEFITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>More flexible architecture</td>
<td>More effective integration with business partners</td>
</tr>
<tr>
<td>Integration of existing applications</td>
<td>Support customer service initiatives</td>
</tr>
<tr>
<td>Improved data integration</td>
<td>Enable employee self service</td>
</tr>
<tr>
<td>Supports business process management</td>
<td>Streamline the supply chain</td>
</tr>
<tr>
<td>Facilitates enterprise portal initiatives</td>
<td>More effective use of external service providers</td>
</tr>
<tr>
<td>Speeds custom application development</td>
<td>Facilitate global sourcing</td>
</tr>
</tbody>
</table>

Table 4: SOA Technology and Business Benefits
Table 5: Mapping SOA and EA Architecture Domains [23]

<table>
<thead>
<tr>
<th>Architecture domains</th>
<th>SOA solution stack</th>
<th>EA framework</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business</td>
<td>Business process</td>
<td>Business architecture</td>
</tr>
<tr>
<td>Applications</td>
<td>Services and components</td>
<td>Application architecture</td>
</tr>
<tr>
<td>Integration &amp; Middleware</td>
<td>Integration architecture / ESB</td>
<td>Technology architecture</td>
</tr>
<tr>
<td>Data</td>
<td>Data architecture</td>
<td>Information architecture</td>
</tr>
<tr>
<td>Operations</td>
<td>QoS, security, monitoring, and infrastructure</td>
<td>Technology architecture</td>
</tr>
</tbody>
</table>

• Both address similar architectural domains.
• Both are intended to closely align IT with business.
• Both use input based on business objectives.
• Both require similar strategies and planning activities.

• EA focuses on defining business components, while SOA focuses on business services.
• EA deals with application frameworks and enterprise applications, while SOA’s scope is on service modelling only.
• EA deals with enterprise-level infrastructure including servers, databases, and so on, while SOA focuses on the infrastructure that supports services, namely the Enterprise Service Bus.
• EA addresses enterprise integration patterns and when they should be used, including point-to-point integration; file transfer, and other traditional application integration approaches. SOA provides an integration approach based on using services. Though the SOA approach to integration may prove to be the most flexible and recommended approach, you should consider it as one of the approaches EA needs to define and support.

Table 6: Complimentary activities in SOA and EA Interactions

![Figure 1: SOA Components and Relationships adapted from [4]](image-url)
ACKNOWLEDGMENTS

The author would like to acknowledge the assistance and support of both the CSIR and ABSA in terms of sponsorship for the conference.

REFERENCES


