Enterprise Resource Planning (ERP) System Development Using Service Oriented Architecture

SOA-Based Erp System Development

Salisu Garba
Department of Mathematics and Computer Science
Sule Lamido University
Kafin Hausa, Nigeria

***Abstract*—Despite the substantial financial commitment budgeted for Enterprise Resource Planning (ERP) systems; organizations continue to express dissatisfaction with the flexibility, reusability and agility of ERP systems. The changes that are taking place in business and technology today are accelerating with increase in competition putting pressure on organizations and management. To enhance business performance, organizations need an effectual planning and control system that synchronize all business procedures across the organization and adopt both technological and business changes. This paper present a theory-grounded SOA-based ERP system development framework. One of the potential strengths of the framework is the ability to deploy ERP systems in form of Software as a Service (SaaS) which can easily be deployed in the cloud to enable the concept of IT as a service that can solve issues such as application integration over heterogeneous platforms, protocols, and devices.**

Keywords*— *Aluminium ERP system, Service Oriented Architecture, SOA-based ERP system, SOA-based Framework.

I. INTRODUCTION

Enterprise Resource Planning (ERP) originates from material requirement planning (MRP) then later manufacturing resource planning and has evolved into software packages that coordinate different business functions. Its visibility can be traced back to the mid-1990s, and it gained momentum in the mid-2000s.

Ragowsky, A., & Gefen, D. (2008) define ERP Software as a multi-module information system that integrates enterprise-wide business functions and automate business activities aiming to facilitate information flow between all business functions. Basu, V., & Lederer, A. L. (2011) argue that ERP is not a system, but a framework that unites major business processes within a single family of software module.

ERP integrates enterprise-wide external and internal management information through coordinating accounting, sales and marketing, services, customer relationship management (CRM), manufacturing, and to facilitate business planning, information sharing, and decision making across an entire organization.

Components of ERP system include transactional database, business intellect system, customizable reporting, external admission via technology such as web services, exploration functions, document management, messaging/chat/wiki, workflow management. The fundamental objective of ERP system is to make business better; therefore, to fully understand the technical aspects of an ERP system it is significant to understand the business aspects of ERP system (ERP Reference Architecture).

ERP systems are typically architecture in a modular fashion with different modules each focusing on different business functional area. This allows for automating some business functions but not necessary all in an organization. Common modules for example finance and accounting modules are adopted in nearly all implementation. But a service organization like Informatics Academy do not need manufacturing module. This ERP architecture is advantageous for heterogeneous clients.

According to Monk, E., & Wagner, B. (2012), the majority of ERP systems have four functional areas (modules) which are Accounting and Finance, Human Resources, Marketing and Sales, and Supply Chain Management.

Enterprise resource planning (ERP) systems have proven to have visible benefits in business integration, strategic planning, operational cost reduction, business flexibility and much more, that can be seen by many organizations. But the architecture used in developing these systems does not offer the contemporary services needed for uniting the best features of on-premises application (for example, low latency and affluent functionality) alongside the best features of cloud computing (for example, elastic scalability and flexibility). Successful implementations of ERP system increase performance in all the three levels of organization that is Operational control, Management control, and Strategic planning. ERP can improve the efficiency of organization through automation, integration, and sharing of real-time information.

II. SERVICE ORIENTED ARCHITECTURE (SOA)

Service Oriented Architecture (SOA) is widely used but the term lacks precision, therefore SOA means disparate things to disparate people. From a software engineering outlook, Kontogiannis, K. et al. (2008) define SOA as a software architecture based on the delivery of reusable, well-defined business services that are supported by IT components in such a way that the business services are loosely coupled to greatly minimize the impact of change.

Prior to Service-oriented architecture, the Common Object Request Broker Architecture (CORBA) and the Distributed Component Object Model (DCOM) provide similar and related

functionality. These existing approaches to service orientation, however, suffered from a few tricky problems such as tightly coupled scenarios.

It is significant to recognize that SOA is not a technology, but a method of software design that propose a fundamental shift in how organizations implement business systems with intend in changes of technology, methodology, and organizational structure. Papazoglou, M. P., & Van Den Heuvel, W. J. (2007) points out that SOA mark the end of monolithic enterprise applications and mark the commencement of more flexible and adoptable business process centric application.

Service Oriented Architecture (SOA) is devise to standardize obtainable IT resources and transformed the heterogeneous collection of distributed, intricate systems and applications into a set-up of integrated, straightforward and flexible IT assets.

Seth, A. et al. (2011) suggest that to avoid weaken organizational agility; IT assets have to flexibly respond to changing business needs. SOA support platform independence, technology independence and language independent to provide greater flexibility for assembling new solutions from existing IT components regardless of located or source of the components.

Service Oriented Architecture offers design framework to integrate siloed systems so that the functionality of those systems can be access as services on a network. Mainly implemented via standards-based web services, SOA split down monolithic systems into a collection of services, implement functionality in a modular fashion.

A service is an implementation of well-defined company functionality, consumed by clients in disparate applications or company procedures. Mike Papazoglou, M. P., & Van Den Heuvel, W. J. (2007) mentioned that Services are connect together using Web Services. However, Web services were merely a step along a much longer road. Web Services are the composition of protocols by which Services can be published, discovered and utilized in a technology impartial, methodology neutral, platform neutral, and language neutral standard form.

The core concept of SOA is service, however the relationship between the three functions of middleware are equally important (publish, bind and find). A service can be published for users to find, after the user go through service access control protocols such as authentication.

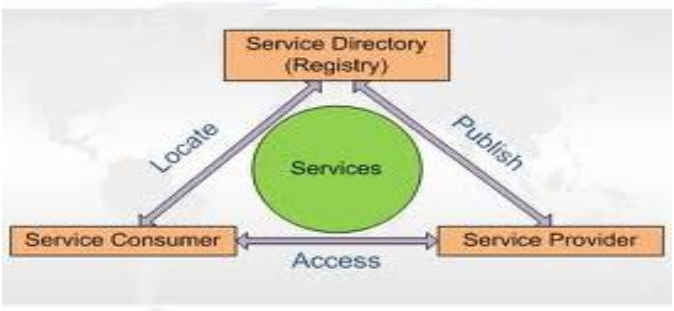


Fig. 1. The Basic SOA Architecture.

III. SOA Rationale and Components

A considerable amount of literature has been published on the SOA rationales and components. Enterprise architects regard SOA as an architectural evolution rather than revolution as it captures many of the excellent features of previous software architectures. Services are the building blocks of any software architecture, which is the implementation of well-defined business functionality, consumed by clients in different applications or business processes.

Seth, A.et al., 2011) identify the key driving forces behind the need and adoption of SOA in enterprise application development.

- Failure of IT to respond to the changing business needs of the enterprise
- Cost and complexity of IT and of inter-organizational integration.
- Problems with the current best of breed application silo architecture.
- Need to create and support intra and inter-enterprise business process.
- Need to standardized technology infrastructure and adopt new standard-based technologies.

A recent survey by Murugesupillai, E. et al., (2011) highlighted some of the major rationale of SOA that can address the issues of ERP systems. Sixty one percent (61%) of the respondents cite quick and cost effective responds to change in technology, methodology and organizational structure. Twenty five percent (25%) of the respondents cite more flexible integration and reconfiguration of business processes. Fourteen percent (14%) of the respondents cite Cost of managing IT and organizational resources.

In a large longitudinal study by Ramollari, E. et al. (2011) shows that 67% of large enterprises are considering or pursuing SOA to improve effectiveness and efficiency of the underlying business process. Constant improvement in technology enable innovation that attract strategic benefits, SOA helps in the adaptation of future business changes in terms of technology upgrade, system customization and maintenance. Eighty six percent (86.67%) of the surveyed SOA experts cited both easier application integration and easier introduction of new services as major motivational factors of SOA in enterprise application development.

Austvold, E., & Carter, K. (2005) conducted an industrial survey in which shows that SOA cost 20% less to implement and save 50% more with each reuse than traditional component-based development. Even though it is hard-to-quantify, increases in productivity with averaged 2.5 times more than non-SOA development.

One of the essential diversions between Object Orientation or Component Orientation and SOA is the existence of policy. Ramollari, E. et al. (2011) demonstrated that Policy split dynamic specifications from static/semantic specifications that are externalized from the business logic and can be updated in run-time. The security properties such encryption, authentication, are determined by the policy. The components of SOA enable plug and play services to support business

processes via a standard interface. Figure below illustrate the relations between SOA components.

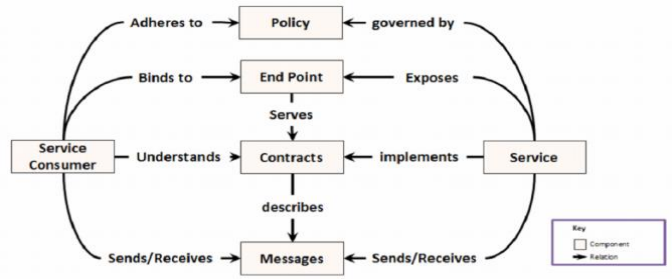


Fig. 2. SOA components.

IV. SOA Design Principles

A considerable amount of literature has been published on SOA design pattern (Arsanjani et al., 2009; Seth, A. et al., 2011). These studies have describe the ten most widely used design patterns, which are Agnostic Services, Agnostic Service Declaration, Atomic Service Transaction, Enterprise Service Bus (ESB), Service Façade, Service Callback, Multiple Service Contracts, Authentication Broker, Message Origin Authentication, Message Screening. These design patterns provide a solution to the concurrent problems while implementing the SOA design principles.

Dustdar, S., & Schreiner, W. (2005) defined eight specific service-orientation principles common to all primary SOA platforms. While other authors such as Kontogiannis, K. et al., 2008; take account of Service normalization, Service optimization, Service relevance, Service encapsulation, Service location transparency as other principles of SOA.

The following are ground rules that must be followed in using SOA to develop enterprise application.

A. *Standardized service contract*

The concept of standardized service contract requires Functional Expression Standardization, Data Model Standardization, and Policy Standardization of service contract to standardized way of expressing service capabilities, and reduces data transformation between services.

B. *Service Loose Coupling*

The concept of loose coupling requires that the service contract act as an interface through which service consumers communicate with the service logic and vice-versa in favour of interoperability and technology independence.

C. *Service Abstraction*

The concept of service abstraction requires that the information published in a service contract be restricted to successfully utilize the service, which can be logic abstraction, technology information abstraction, functional abstraction etc.

D. *Service reusability*

The concept of service reusability is to design a well-factored service interface with reuse in mind by anticipating usage scenarios to avoid functional duplication and loss of architectural integrity over time according to (Arsan Jani et al., 2009).

E. *Service autonomy*

The concept of service autonomy requires that services have maximum control over its functionality, the logic or the execution environment to guarantee reliability and reusability. This can be design-time autonomy or runtime to provide strategy for scheming self-sufficient services so that the consequential services are more predictable and dependable.

F. *Service granularity*

The concept of service granularity requires that the business functionality in a service operation reach the optimal scope to provide a cluster of interrelated functionality in a single service operation and strike balance between fine-grained service operation and coarse-grained service operation.

G. *Service statelessness*

The concept of statelessness requires that every operation is functionally isolated enabling concurrent service requests through pooling and sharing of service instances (client-service independence) to get better scalability (Arsan Jani et al., 2009).

H. *Service discoverability*

The concept of service discoverability states that the use of service directly relates to how discoverable the service is. Service discoverability is a design principle, which emphasize on adding interpretable meta-data to increase service reuse and decrease the chance of developing services that overlap in function.

In 2005, Austvold, E., & Carter, K. conduct a survey to determine the standard components used in SOA, The finding shows the most essential and the least essential technology components of SOA that can be used to solve the architectural problems in enterprise applications development as shown in the figure below.

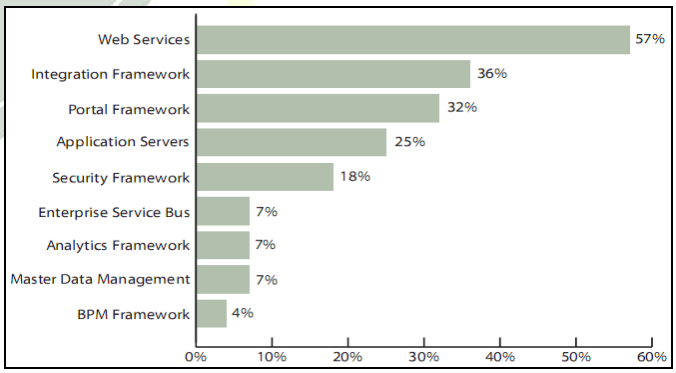


Fig. 3. Most standardized components of SOA.

V. Benifites of SOA

To understand the benefits of SOA abundantly, it is relevant to realize the fit falls of other architectures used in developing enterprise systems. Seeing that enterprise system architecture has implications such as managerial, business and social, therefore it is more than just a technical blueprint in which the implications are restricted to technicality.

The benefits of SOA in developing enterprise applications can be apprehend at two different levels (1) The IT Perspective

(2) Business Perspective. From the IT viewpoint, SOA provides the following benefits.

A. More prolific, extra flexible applications

The SOA methods allows IT to compose in-house IT resources more productive and cost-effective to the business and facilitate the development of a new generation systems that offer different functionality to the organizations regardless of the fundamental technology, platforms and languages for greater flexibility.

B. Faster, more cost-effective application development

SOA principles that promote loose coupling, standards-based technologies, and coarse-grain service design enables the creation of reusable services repository that can be pooled into higher-level services and composite system as new business needs arise. These lower the cost development, testing and maintenance.

C. More manageable and secure applications

SOA provide a common infrastructure for developing secure and conventional services. As business requirements change, SOA allows easier services addition. Furthermore, SOA provides the resources for shielding active IT investments since services are accessed not the applications themselves. Strong authentication and authorization model is used to protect all services given that services exist independently therefore the SOA approach provides greater overall security.

From the business viewpoint, SOA facilitate development of dynamic applications that deal with top-level business problems and concern that are vital to organizational development and competitiveness. SOA solutions support the following areas.

D. Enhanced business decision-making

Aggregating business information from multiple peoples, multiple processes, and multiple departments into a dynamic, composite business application, mapped using a single unified view, allows organizations to make a sound decision and react quickly to anticipated problems.

E. Greater employee productivity

The agility of SOA allows business process improvement; enabling employees to attend other semi-structured activities, rather than conforming to the restrictions, narrow constrains of the underlying IT systems. Furthermore, users can access information using different device, in different language which in-turn enhanced employee productivity.

F. Stronger relations with suppliers and customers

Managing the connection between any organization and external parties is very strong point for business prosperity. The benefits of SOA broaden beyond organizational margins. Integrating disparate systems and trading partners allows collaboration that unlocks critical supply and demand chain processes thereby enabling better alignment of processes with organizational strategy.

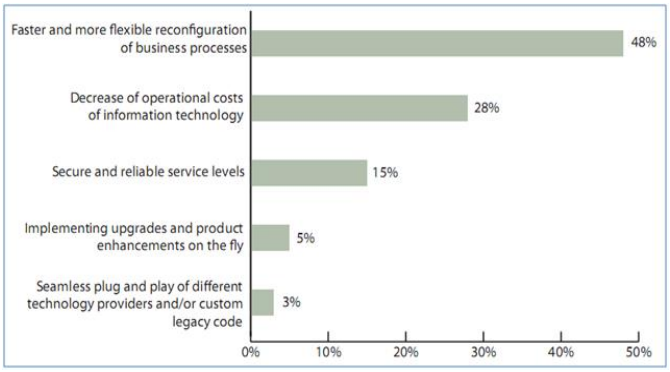


Fig. 4. The Expected Benefits of SOA.

Another recent study by Murugesupillai, E. et al., (2011) have examine the benefit of SOA, which further support the above discursions with the following statistical results.

- Improve Flexibility (97%)
- Decrease Cost (86%)
- Reduce Risk (71%)
- Increase Revenue (51%)
- Enable new Product (43%).

VI. PROPOSED SOA FRAMEWORK

After going through considerable amount of literature on the existing Enterprise Software Development frameworks and SOA, the author proposes a SOA-based Enterprise system development framework in this section.

The three layers (Front, Façade, back) are the fundamental elements used for designing the proposed framework, with moderately encrusted of compound services that align with organizational business processes.

The upper part of this framework consists of service consumption process. The middle part of this architecture consists of business logics, service orchestration process that are enfolded, and wrapped in a single façade. The lower part of this architecture consists of data access and data storage processes. Figure16 represents the mapping of SOA with the three layers of Proposed Architecture. The activities in each layer are discussed below.

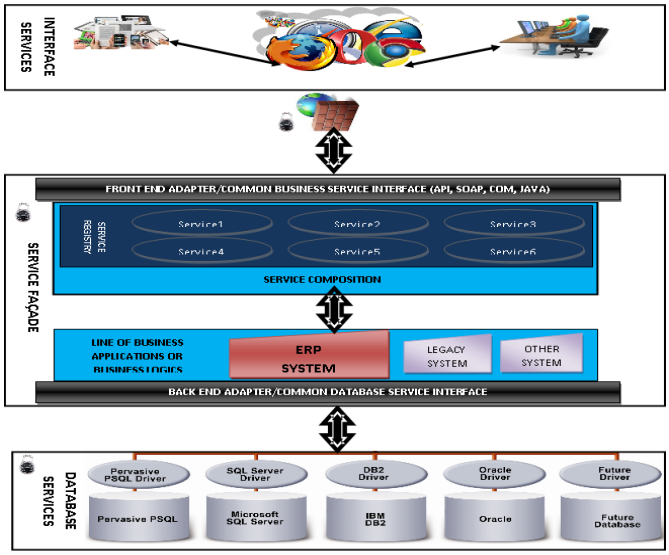


Fig. 5. Layered view of the proposed SOA-based framework.

In the figure above (Figure 5), the services such as web services that are equivalent to business objectives or organizational business processes are identified from the Line of Business Applications or Business Logics and are positioned into the service repository using description protocol, specifically web service definition language (WSDL).

When a service request is received from the presentation layer, the service façade directs the request to the application layer. The requests are processed according to the priority and the availability of the requested services. On process completion, the processing result is sent via the service façade back to the presentation layer. The GUI (usually the browser) interprets the result for user's viewing.

These web services are composition or orchestrated actively to portray business process. The structural designs of this framework itself is ad-hoc in nature and if at any time, there is a change in business objectives or organizational business processes, the services are intelligently orchestrated actively to act in accordance with new business process.

The proposed framework shows how the service registry can be accessed to discover old or new services by sending a message through the service façade using uniform resource identifier (URI). These messages are sent/received repeatedly until a certain goal of business process is reached, for example, placing order followed by an acknowledgement, an invoice, payment notice, etc.

The system security can be attained through encrypting some of the sensitive messages and single user login authentication.

The composite services are formed when other web services (business unit specific components, project-specific components, enterprise scale components etc) are orchestrated to accomplish a business process or task. For example, an order service may need to contact inventory services for pricing or supply chain service may need to access manufacturing service for availability.

The web services are discoverable at front-end adapter/common business service interface (endpoints) before finally reaching the presentation layer. The endpoint manages the translation between the asynchronous world of messaging and the synchronous application program. As separate applications and services use data in different format, which may be incompatible, a translation of the message is often required along the way.

Furthermore, the set of poorly-designed application programming interfaces (API) are wrapped with a single well-designed application programming interfaces (API) for every task to be performed. The dependency between the front-end (presentation layer), back end (database layer) and the services within the façade will be reduced to fulfill the reusability, flexibility and scalability promises of SOA.

This framework is designed to handle multi-database systems. Back-end adapter/common database service interface enables communication with the back-end databases of the enterprise to provide scalability and efficient data management (data storage, access, update, deletion or any other database functions). Reports can be generated from the back-end systems. In this context, an online data backup system can be added at the back-end to prevent data loss in the case of a system failure.

VII. Suitable Technologies for Implementation

The presentation layer of the proposed framework can be design using any web technology but the most suitable are JSP, or ASP.Net. Java Server Pages (JSP) technology uses tags comparable to XML tags that encapsulate the logic that produce the content for the page.

The ERP application logics are place in application server such as J2EE server or .NET server or another server component from which the page accesses the tags. Any of the tag formatting such as XML or HTML are retrieve directly back to the response page. As a result of separating the page logic out of the design and sustaining a reusable component-based design, Java Server Pages (JSP) technology formulated the quicker and uncomplicated technology in building enterprise application interface or presentation layer.

Active Server Pages (ASP) technology enables the creation of dynamic, enterprise application interface or presentation layer using compiled languages such as VB and C#. The ERP application logics are executed in application server such as J2EE server or .NET server or another server component and the outputs are portrayed in the client's web browser. Active Server Pages (ASP) technology allows easy code reusability, deployment, and maintenance together with a better performance.

The service façade of the proposed framework can be design using any technology but the most suitable are J2EE beans, or .Net framework.

The enterprise beans, or application components can be deploy into the container and the container settings customize the underlying support provided by the J2EE server, which includes services such as security, transaction management, Java Naming and Directory Interface (JNDI) lookups, and remote connectivity.

The Service Façade can be called from a standard .NET server component. The ability of .NET to maximize the efficiency of code reveal the same real-world condition as J2EE beans but uses modern, built-in .NET features, such as

reflection, object initializers, delegates, automatic properties, generics etc. The service façade is able to scale the system to sustain a large number of clients at the same time.

The database layer of the proposed framework can access any enterprise database management system but the most suitable are Oracle, Microsoft SQL Server, and DB2.

VIII. RESULTS AND DISSCUSSION

To achieve useful and excellent evaluation results, the following approaches are use.

- The developed SOA-based ERP prototype will be used to conduct UAT (User Acceptance Test) where the users test out the system and give their opinion.
- Comparison Method is used to compare the proposed SOA-based ESD development framework with the existing similar approaches.
- Factor Rating Method is use to identify and evaluate the score of the SOA-based ESD factors.

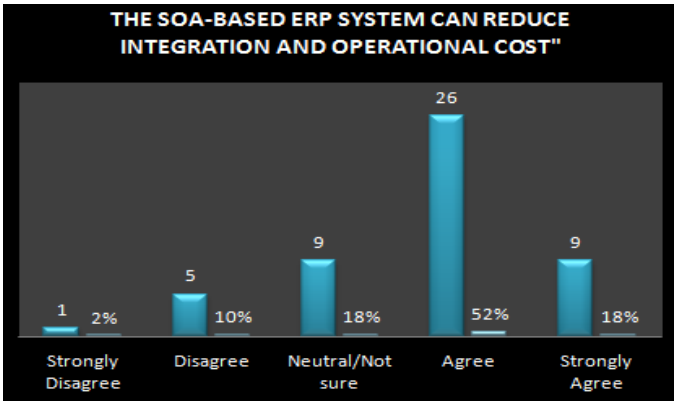
The comparison method evaluates the core technical effectiveness of the proposed SOA-based Enterprise Software development framework with other existing similar approaches. For evaluation, CORBA, XML web services and Web sphere MQ are identified as the direct or indirect related approaches. The table below shows the comprehensive comparative analysis of the proposed SOA-based ERP development framework with different comparable approaches (i.e. CORBA, XML web services, Web sphere MQ).

TABLE I. THE COMPARISON METHOD RESULT

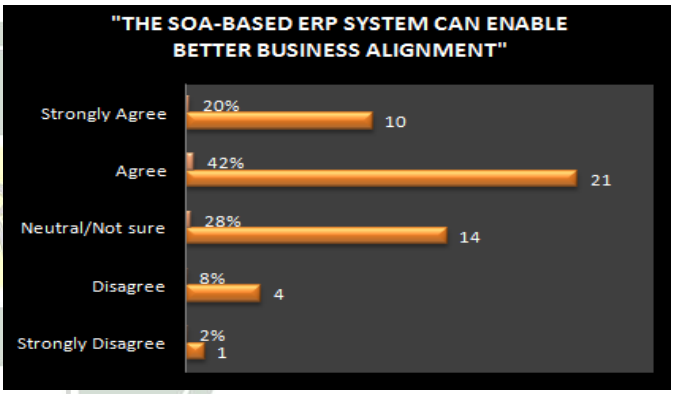
No	Elements of Service platform	Proposed Framework	CORBA	XML-web services	Websphree MQ
1	Open standards	Yes	Yes (OMG)	Yes (Fragmented)	No (Propriety)
2	Loosely coupled	Yes	Mediocre	Yes	Yes
3	Language neutral	Yes	Yes	Yes	Mediocre
4	Technology neutral	High	Moderate	High	Ad hoc
5	Platform neutral	Yes	Yes	Yes	Yes (Weak)
6	Load balancing	Yes (Embed within the Architecture)	Yes	Yes (Weak)	Yes
7	Interface repository	UDDI	CORBA–interface	UDDI based	None
8	Publish- subscribe	Yes (strong Support)	Yes	Yes	Yes (weak)

From the above table, it can be concluded that the proposed SOA-based ERP development framework is capable of reducing the degree of risk and establishing greater flexibility with the ability to adapt to changes more easily and quickly. The number of services developed can be tracked and reused to cut down the substantial financial commitment budgeted for Enterprise Resource Planning (ERP).

Goal, Question, Metrics Method is adopted to devise an experience survey to evaluate the effectiveness of the SOA-based ERP system prototype. In response to the statement, "The SOA-based ERP system can reduce integration and operational cost". The majority of the IT professionals clearly agree with the statement as the below figure shows a staggering 70% of the responses fell within the "agree" to "strongly agree" bands. Even though 18% of the responses are neutral, the mean value (3.74) of the responses proved to be convincing enough.



Similarly, more than 60% of responses fell within the "agree" to "strongly agree" bands in response to the statement "The SOA-based ERP system can enable better business alignment". The mean value is 3.7. The high mean value express a fulfilling result as shown in the figure below.



IT Industry people are asked to evaluate various properties on 5 point scale, where each point has a significance of (Poor, Fair, Good, Very Good, Excellent) and a value of 1 to 5. The table below shows the total score of each factor out of possible 250.

S/N	SOA-BASED ERP SYSTEM FACTORS	MEAN VALUE	MODE VALUE	POINT SCORED
1	The Interface	4.38	5 (Excellent)	219
2	The Functionality	2.73	2 (Fair)	136
3	The Compatibility	3.42	4 (Very Good)	171
4	The Portability	3.16	4 (Very Good)	158
5	The Flexibility	3.82	4 (Very Good)	191
6	The Interoperability	3.26	3 (Good)	163
7	The Reliability	3.2	3 (Good)	160
8	The Reusability	3.44	3 (Good)	172
9	The Usability	2.76	3 (Good)	138
10	The General Performance	2.8	3 (Good)	140

The majority of the responses are either "Good" or "Very Good"; interestingly most of the SOA properties of the system such as flexibility, compatibility, reusability get the higher percentage of the positive responses.

However, the functionality, the usability, and the general performance of the system have the lower mean value and the total point scored, because the system is only a prototype. With all this, any of the evaluation factors that have more than 2.5 mean value and more than 125 points scored proved to be convincing enough.

CONCLUSION

A new SOA-based ERP system development framework was developed. This SOA approach also incorporates other elements that are unique to the ERP systems, such as standards interface technologies that improve the transitions.

With the creation of a SOA-based ERP system development framework, the project is responding to the needs of resolving the current weaknesses and pitfalls of ERP systems in organizations. In addition to that, the proposed SOA-based ERP system development framework also constitutes the foundation for the targeted business adaptability, aligning enterprise IT capabilities with business goals, improved business efficiency and adoption to technological changes.

Seen an encouraging response after analyzing the survey question, it is not wise to jump directly to the conclusion as there are other requirements and future research that must be given due consideration. This is a sample result with a relatively small number of respondents (50), even though the responses can be reliable considering the information technology background of the respondents. The result is not correlated with the age factor or working experience factors and the system is just a prototype.

REFERENCES

[1] Arsanjani, A., Booch, G., Boubez, T., Brown, P. C., & Chappell, D. John deVadoss, Thomas Erl, Nicolai Josuttis Dirk Krafzig, Mark Little, Brian Loesgen, Anne Thomas Manes, Joe McKendrick Steve Ross-Talbot, Stefan Tilkov, Clemens Utschig-Utschig, and Herbjörn Wilhelmsen. The SOA Manifesto, 2009.

[2] Austvold, E., & Carter, K. (2005). Service-oriented architectures: Survey findings on deployment and plans for the future. AMR Research, 0250-1361.

[3] Seth, A., Agarwal, H., & Singla, A. R. (2011). Designing a SOA based model. ACM SIGSOFT Software Engineering Notes, 36(5), 1-7.

[4] Monk, E., & Wagner, B. (2012). Concepts in enterprise resource planning. Cengage Learning.

[5] Ramollari, E., Dranidis, D., & Simons, A. J. (2007, June). A survey of service oriented development methodologies. In The 2nd European Young Researchers Workshop on Service Oriented Computing (Vol. 75).

[6] Murugesupillai, E., Mohabbati, B., & Gašević, D. (2011, August). A preliminary mapping study of approaches bridging software product lines and service-oriented architectures. In Proceedings of the 15th International Software Product Line Conference, Volume 2 (p. 11). ACM.

[7] Kontogiannis, K., Lewis, G. A., & Smith, D. B. (2008, May). A research agenda for service-oriented architecture. In Proceedings of the 2nd international workshop on Systems development in SOA environments (pp. 1-6). ACM.

[8] Papazoglou, M. P., & Van Den Heuvel, W. J. (2007). Service oriented architectures: approaches, technologies and research issues. The VLDB journal, 16(3), 389-415.

[9] Dustdar, S., & Schreiner, W. (2005). A survey on web services composition.International journal of web and grid services, 1(1), 1-30.

[10] Basu, V., & Lederer, A. L. (2011). Agency theory and consultant management in enterprise resource planning systems implementation. ACM SIGMIS Database, 42(3), 10-33.