Technical Analysis for Dynamic Integration of Heterogeneous Services

Makaziwe Makamba, Jabu Mtsweni, Ernest Ketcha Ngassam

***Abstract*—The proliferation of heterogeneous services has accelerated the growth of digital economy. This is stimulated by the increasing number of users who accomplish lot of tasks electronically. In today's digital economy, services are the key components for communication and collaboration amongst enterprises. For heterogeneous services to interact, service integration needs to take place. Thus, the service integration becomes a critical issue in this domain.**

In the real world, services are heterogeneous, as they have been developed by different vendors, using different programming language, architectures, and platforms deployed in different environments. The heterogeneity of these services brings challenges in the integration domain. The shift of technology has forced many service enterprises to develop integration techniques that would be able to support the changes in services and enable integration of heterogonous services. Service integration is not new phenomenal, however the previous and current integration techniques uses static method of integrating services such as an adaptor models, point-to-point, middleware's, enterprise service bus (ESB) and other integration techniques. These techniques uses static methods of integrating heterogeneous service, which makes these integration techniques expensive, prone to human errors, difficult to maintain and unable to adapt with dynamic changes of services.

In this paper, we focus on analyzing the technical challenges of integrating heterogeneous services using existing approaches, with an aim of recommending a novel technique that could eliminate these service integration challenges and provide a dynamic method of integrating heterogeneous services with a minimum user-involvement. This paper provides a technical analysis of how these current integration techniques operate and what needs to be done to develop a technique that would enable dynamic integration of heterogeneous services with a minimum user-involvement.

***Index Terms*—Dynamic-Integration, Heterogeneous-service, Cloud-services and Integration.**

Manuscript received June, 25 2017; revised August 05, 2017. This work is supported by Council for Scientific and Industrial Research (CSIR) under the unit of Defense, Peace, Safety and Security (DPSS).

M. Makamba is a PhD candidate at the University of South Africa, Her research interest includes e-services, integration, ICT4D and service integration. She is currently with the Council for Scientific and Industrial Research, Meiring Naude Road, South Africa, 0184(email: Makamba.makaziwe93@gmail.com).

J.S. Mtsweni is a Research Group Leader in the Cyber Defence research domain at the Council for Scientific and Industrial Research. Council for Scientific and Industrial Research, Meiring Naude Road, South Africa, 0184(email: mtswenij@gmial.com). He has published over 50 peer-reviewed conference and journal papers and is well cited locally and internationally. He has delivered a number of technical talks at various conference across the globe.

E.K Ngassam is a Professor Extraordinaire at the UNISA School of Computing. His research interest includes Service Engineering and Service Science, Technologies for Emerging Economies, Information Security, and High Performance Pattern Matching. (email: eketcha@gmail.com)

I. Introduction

The effective communication between service enterprises is critical for their success. Currently, service enterprises require a global integrated system, which can provide the unified information and meet business changes such as organization re-engineering, enterprise emergence, services changes and user requirements [1]. In the contemporary environment, the shift of technologies requires a flexible modernized technique that would stand and support the changes of enterprise services paradigm. Currently, business enterprises are facing challenges of heterogeneous service integration, where services and technology are frequently shifting and user requirements change with a rapid speed [1].

Due to the shift in the development of technology, customers and business enterprises communicate using different services. These services need to interact with each other to achieve the business objectives [2].

Services are described as autonomous software computational components that can be described, published, discovered, orchestrated and programmed for the purpose of developing massively distributed interoperable systems and achieve a business objective [3]. For services to interact, it is essential to have a technique that allow these services to be integrated in a seamless fashion.

Service enterprises and developers across the globe have developed and implemented various integration techniques such as, point-to-point (P2P), spoke and hub, middleware's and Enterprise Service Bus (ESB) that allow services to be integrated [3]. Since service integration depends on different business requirements, customer demands, services, and resources, different integration solutions are compulsory, due to the fact that a single integration solution seldom satisfies all requirements. Due to the heterogeneity of services today, service integration has become a critical challenges in the domain of integration.

In this context, a heterogeneous service is defined as a distinctiveness of a service that is developed on different environments using different techniques and architectures to complete a specific purpose [3].

The challenge rises when there is a need to integrate these heterogeneous services as they are built for different purposes and uses different architectures, platforms and different programing languages. The current service integration techniques require a massive user-intervention when integrating heterogeneous services and do not accommodate dynamic integration of these services [3]. Service integration is a key for the success of enterprise systems (ES) [4].

In this paper, we analyse the existing challenges of the

ISBN: 978-988-14047-5-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

current service integration techniques. These techniques are further compared and contrasted following a pre-determined criteria for effective dynamic integration of heterogeneous services. These existing techniques are explored with an aim of designing and developing a technical model that could enable dynamic integration of heterogeneous services with a minimum user-intervention.

The outline of this paper is as follows: Section 2, provides details about service integration. In Section 3, this paper explains the current integration techniques and stipulates the advantages and limitations of these techniques. Section 4, provides explicit details about the challenges of integrating heterogeneous services. Section 5, discusses the work that has been done by other researchers and service enterprises to reduce this challenge of heterogeneous service integration.

Section 6, gives recommendation on the technique that needs to be developed to enable dynamic integration of heterogeneous service with a minimum user-involvement. Section 7, concludes the paper.

II. Service integration

Service Integration is a critical component in the integration arena, as it enables the streamlining of e-service domination, while simultaneously bringing tangible business benefits; such as reduced operating costs, decreased risk, enhanced governance and compliance, and improved service quality [4]. The magnitude of online services that needs to interact at the same time and meet different user requirements, have rapidly increased with the colossal technology development. It is therefore essential for these services to achieve semantic interoperability among the information exchange entities and be able to dynamically integrate with each other.

At present, an increasing number of proprietary business processes, heterogeneous data standards, and diverse user requirements have made it difficult for services to be implemented using adaptable, extensible, and scalable technologies. With the connectivity of Web, enterprises of any size can interact with many other enterprises and/or consumers anywhere in the world. It is evident that information exchange plays a key role in these interactions. Hence, the benefit of this global reach can be realized only when the information is exchanged efficiently and meaningfully.

Many organisations and researchers across the globe have realised that service enterprises seek to procure best-in-class solutions for each of the service domains to leverage the services integration [5]. This is intensified by the integration technology problems. Many organizations find themselves paying excessive amounts to many different services providers who can assist in integrating legacy services in a flexible and scalable manner [5].

It is evident that any viable solution to this problem has to be adaptable to accommodate changes, extensible enough to allow easy addition and removal of services, scalable enough to handle a large number of data sources efficiently, support interoperability and it should require a minimum user-intervention [6]. To examine the current predicaments of many service enterprises, we scrutinize the current service integration techniques, with the aim of analysing the gaps and challenges that need to be addressed in order to solve the service integration challenges in the area of integration, particularly when dealing with heterogeneous services.

The following section elaborates more on the current service integration techniques that have been developed in an attempt to address the challenges pertinent to heterogeneous service integration.

III. Integration Techniques

The progressively volatile economic paradigm coupled with the proliferation of technology and service platforms, has maximized the integration ramifications. Many service enterprises are gradually striving to be more agile and flexible, and they expect service integrators and developers to facilitate the integration of services, since the traditional integration models lack flexibility and require intense manual user-involvement to integrate services [6].

As service enterprises move from small scale service registries to large scale service repositories, there is an increase in heterogeneity of services with respect to their service providers, their interface description languages (IDLs), and their offered quality of service (QoS). As the need to integrate heterogeneous services is increasing, there is a necessity to provide a dynamic and sustainable service integration technique.

Any sustainable service integration technique has to be dynamic, simple, extensible, adaptable, interoperable, and scalable, accommodate heterogeneity of services and involve a minimum user-intervention when integrating services. These components are the key criteria that determines an effective and sustainable dynamic integration technique. These key integration criteria are used in this paper for the comparison of these current integration techniques to stipulate benefits, limitations of the existing integration techniques and illustrate the gaps that need to be bridged in this domain.

In the current integration techniques, a developer has the difficult task of predicting all of the possible ways that the users will want a given service to work with all other diverse services. The current integration techniques use the static service integration method, where services are integrated at the design time, which requires a developer's involvement wherever there are changes on a service, current techniques do no support heterogeneity of service and dynamic integration where services can update themselves whenever there are challenges [3].

The limitations of these current integration techniques impact the developers/ service integrator, service providers and service enterprises [7].

It is essential that a sustainable service integration technique has to consolidates, validates and simplifies the services integration, since the current integration techniques have limitations that hinder the production of service enterprises.

In section 5, this paper compares and contrasts the existing integration approaches, to identify the components that needs to improvement in this area of integration, using the integration criteria that is mentioned in this section. However, before we compare the current technical approaches, it is essential to discuss the current challenges

ISBN: 978-988-14047-5-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

that face service enterprises and developers across the globe.

IV. CHALLENGES OF HETEROGENEOUS SERVICE INTEGRATION

The Internet is evolving not only to provide information and e-commerce transactions, but also to act as the platform through which services are delivered to businesses and customers. These electronic services or e-services could become a key part of the value provided by many businesses [8]. However, while organizations have define syntax to solve their specific problems, the multitude of syntax (schemas) create incompatibility problems with schemas developed by others.

According to the current literature, heterogeneous service integration has become a problem due to the diversity of technological platforms, frequent service changes and the current integration techniques that do not meet the required standard of integrating heterogeneous services [7].

This has led to many organizations to employ different integration techniques, which are expensive, difficult to maintain and require intense code to maintain the integration or update integration protocols [8].

Most of the organizations integrate services using the traditional approaches where a connector needs to be implemented whenever a services have to be integrated. Some enterprises realize the expense of integrating service using vendors, as that results to vendor-lock in [9]. The service integration process is not an easy task and it is one of the main research issues in this area.

Currently, services are integrated in static manner which creates challenges whenever there are changes in services, as this requires an intense user-involvement when the services needs to be integrated. This makes integration difficult especially when heterogeneous services needs to be integrated, the lack of dynamic integration creates challenges in service integration area. However, the current literature proves that there is no infrastructure or tools available for facilitating the dynamic integration of heterogeneous services and interoperability of such services with a minimal user-intervention.

V. RELATED WORK

The current literature has stipulated that the key solution to this service integration challenge is a technique that can enable dynamic integration of heterogeneous services, where services can be integrated and disintegrated after the work is done [10].

In this paper, we identified a number integration techniques that are utilized in the service enterprise domain and their shortcomings. Due to the challenges of service integration, many enterprises have developed and implemented the point-to-point integration technique.

Point-to-point (P2P) is an integration technique that uses a direct communication channel established between two services [11]. Point to point integration technique has been opted by many enterprises in previous decades as a main technique to facilitate service integration [11]. Due to the benefits of this technique, many small enterprises utilized this approach [12]. However, due to its limitations when there are many services that need to be integrated, many service enterprises discarded this approach [11]. The following Figure1, demonstrates how p2p operates and disadvantages of using this technique.

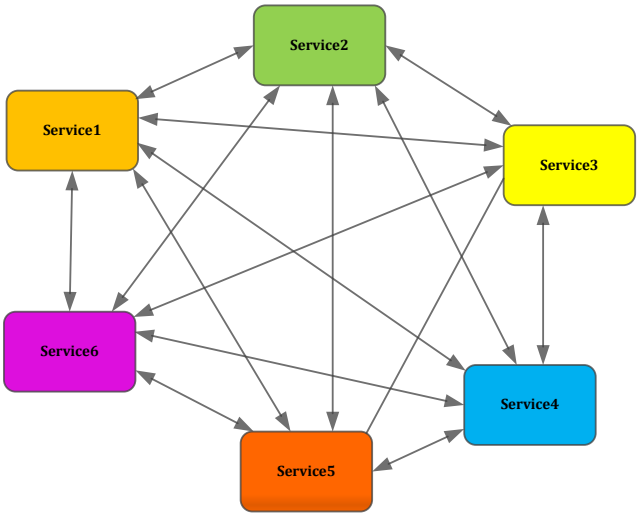

Figure 1: Point-to-Point Integration Technique

This type of integration technique uses a static method of integration where services are manually integrated at the design time. The challenges of using this technique is an increasing number of interfaces that needs to be creates when integrating services. As depicted on the figure 1 above, when integrating services using this technique, a developer needs to establish an interface for each service which makes this integration approach difficult to manage due to multiple interfaces. This technique does not support interoperability of services and the dynamic integration, since services are integrated at design time.

It was also discovered that this technique uses tight-coupling where services need to know all the information about the service that it needs to integrate with. Due to the fact that this technique uses tight-coupling and transmission control protocol/Internet protocol (TCP IP), for services to be integrated they need to be available on a network at same time. It is excruciating to use this kind of integration technique in a large and busy service enterprise where many services needs to be integrate at a fast speed.

This integration technique does not accommodate and support the integration of heterogeneous services. To make this clear, we measure this technique against the pre-determined integration criteria that is highlighted in Section III of this paper and further used below to analyse the existing integration techniques.

Adaptability - P2P is not adaptable as there are lots of configurations that need to take place whenever there is a new service that needs to be integrated. Each service needs to know all the details about the service it needs to connect with. [12].

Scalability - In terms of scalability, P2P is one technique that does not support scalability, it becomes more unmanageable when there are many services that need to be integrated.

User-Intervention- This integration technique requires a severe user-involvement when integrating services. For example whenever there are changes on a service, an

ISBN: 978-988-14047-5-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

adaptor needs to be updated so that other services can know all the updated details [12].

Interoperability-This technique does not support interoperability as it mostly uses tight-coupling.

Extensibility- Extensibility is not supported when using P2P integration technique, due to the complexity this technique experiences when more services are added.

Simplicity - As mentioned in the table 1 below, P2P is simple to use when integrating few services, however it becomes the most complex technique when there are many services that need to be integrated.

Due to these of shortcomings of this integration technique, service enterprises have opted to use Spoke and Hub integration technique. Spoke and Hub is a static integration technique that integrates services through a central message broker [13]. Spoke is defined as an adapter that connects services to the hub. This technique operates by setting up a centralised server where all the service data gets passed on to multiple service destinations [13]. The central service controls the message flow. This technique operates by using a script to manage adapters, data and IP addresses. The script is a set of commands that are developed to manage data flow and reconfiguration of adapters [14].

The following figure 2 illustrates how services are being integrated when using spoke and hub integration technique.

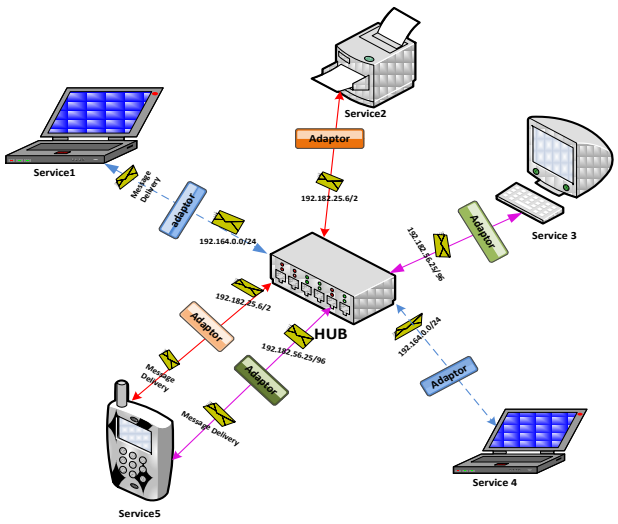

Figure 2: Spokes and Hub Integration technique

As illustrated in the figure 2 above, this technique uses a central hub to integrate these heterogeneous services. This technique offers more advantages than the previous integration technique, which is P2P. However, this technique has its limitations that makes it challenging for service enterprises to use this technique, such as single point of failure, manual configuration of adapters whenever a service needs to integrate with another service. Due to the single hub used in a centre, this technique encounters challenges when there are many services that needs to be integrated at a same time which can result to packets loss. This technique also requires an intense user-involvement when integrating service which can be susceptible to human errors. However, we have measured this technique against the pre-determined integration criteria to evaluate the effectiveness of this integration when integrating heterogeneous services.

Adaptability: This integration technique is limited to the adaptability since a service provider needs to adhere to the certain standards that has been set by the hub in order for service to be integrated.

Scalability: The scalability of this technique is limited depending on the box that is used in in the hosting hub.

Simplicity: Spoke and Hub technique is easy to use and understand.

Extensibility: This technique is not extensible as most of the hubs cannot handle services that need to be integrated at the same time. It cannot manage integration events involving multiple sources and destinations.

Interoperability: This technique is interoperable as it offers reusability of services.

User-Involvement: Like other static integration techniques, this technique requires user-involvement whenever integrating a new service. This means that a code needs to be configured to accommodate the new services.

Due to the challenges of the Spoke and Hub integration technique, Middleware integration technique was developed to overcome the challenges of P2P and spoke and hub integration techniques, this technique was developed to make service integration easy and flexible [15].

Middleware is an integration technique that functions as a piece of software that connects two or more services, allowing them to exchange data [16]. This integration technique uses a static method of integration to integrate heterogeneous services. This technique has been utilized in many service enterprise due to its advantages [16].

Middleware uses adaptor models to integrate services into the middleware. In this technique, a service exposes its interface in order to for a service to be integrated into the middleware. Figure 3 demonstrates how middleware integrates heterogeneous services.

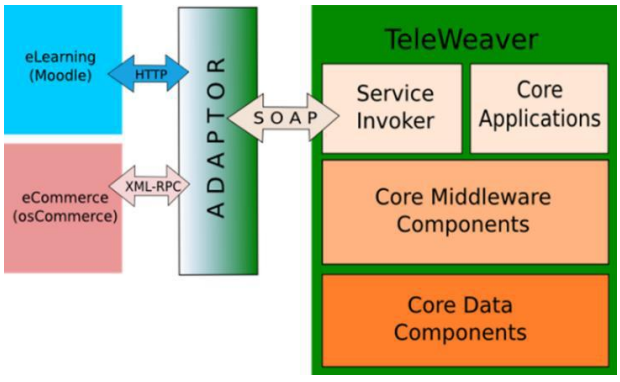

Figure 3: Middleware Integration Technique

As depicted in the figure 3 above, this technique requires an adaptor to integrate heterogeneous services. This technique offers more benefits as discussed table 1 of this paper. This technique has limitations as per our comparative analysis using the set out integration criteria.

Adaptability: This technique has limited adaptability due the fact that it supports certain adaptors that connect services.

ISBN: 978-988-14047-5-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Scalability: This technique promotes scalability as it allows different services to be added and removed from the middleware anytime.
Simplicity: This approach is simple to use, however, it is very complex to configure and maintain.
Interoperability: Middleware integration technique support interoperability.
Extensibility: This integration technique have challenges when it comes to extensibility as there are certain application programming interfaces (APIs) that are supported by the middleware, depending on the development language that a service uses, it might not be supported by the middleware [16].
User-Involvement: This technique requires intense user-involvement, when there is a service that needs to be integrated, a developer need to configure an interface that would supported by the middleware. Also due to the fact that it uses static integration technique to integrate services and that results to a need for a user-involvement when changing services or updating a service in the middleware [17].

Since, there is no one solution to service integration, many service enterprises have opted to use the Enterprise Service Bus (ESB) to integrate heterogeneous services. This technique has been voted as a key catalyst in service integration domain, due to its ability to enable heterogeneous services and due to the fact that it supports most of the key integration criteria [17].

An ESB is an integration technique that combines web services, message routing, data transformation and service virtualization to allow integration of multiple heterogeneous services [18]. In a service integration domain, ESB offers many advantages than other integration techniques. Many service enterprises are still using this technique to integrate their services [19].
However, this technique has failed to support dynamic integration due to its use of a static approach of integrating heterogeneous services. The service implementation is based on message routing engines that support only static message routing for service communication [19]. This brings a challenge in an integration domain where services are dynamically created, updated and deployed. This technique still uses the P2P method of integrating heterogeneous service, in this paper we analyse this technique against the determined integration criteria.
Adaptability: ESB is adaptable as it allows services to be integrated regardless of the development platform, architecture and the language that the services use.
Scalability: the ESB technique offers better scalability compare to point-to-point and spokes and hub integration techniques**.**
Simplicity: This technique is easy to use and simple, due to the reusability and loosely coupled services.
Extensibility: It is extensible as it allows both heterogeneous and homogeneous services to be added to the bus regardless of the development platform.
User-Intervention: To integrate services into the ESB integration technique, point-to-point method is still required, also it uses static integration technique to integrate services.
Interoperability: ESB is a technique that support interoperability and loosely coupled, services can be reusable and interoperable. To illustrate the need for a new technique that could enable dynamic integration of heterogeneous services, Table 1 summarizes the benefits and limitations of the existing integration techniques.

VI. Recommendation

It is evident that the current integration techniques are unable to address the challenges of dynamic integration of heterogeneous services. These services have abilities to address special needs and can be applied more rapidly than traditional service models. However, to sustain the integration of heterogeneous services a strong operational and viable supremacy model that facilitates heterogeneous service integration is essential.

In multi-sourced operating models, where services are provided by a myriad of organizations, ensuring seamless delivery presents a challenge. Thus the ability to efficiently and effectively select and integrate inter-organizational and heterogeneous services over the Web at run-time is important. The recent developments of technologies and standardized services delivery models are driving an influx of distinctive services to an operating environment to both increase capability and reduce cost [20].

The requirements to address service integration in dynamic environments demands a high degree of scalability, context-awareness, translation, mapping, discovery, orchestration and service select while it support interoperability, extensibility, simplicity, adaptability, capability management and it should require minimum user-involvement. To resolve the challenge of heterogeneous service integration, it is important to develop a dynamic integration technique that accommodates all these mentioned criteria and integrate services at run-time. In the near future, we aim to develop the dynamic integration technique that would accommodate all these mentioned criteria.

VII. Conclusion

There is a strong need for dynamic integration technique that would tackle, and ultimately streamline this complex issue of integrating heterogeneous services. Integrating heterogeneous services does not only bring relief to the developer who is responsible for integrate these services, but it also adds value to service enterprises. The added value is derived from the volume and quality of performance and utilization that is made available. It is essential for a sustainable dynamic service integration technique to consolidates, validates and simplifies interoperability and agility of services.

Acknowledgment

The University of South Africa is hereby acknowledged, this work is taken under Council of Science and Industrial Research (CSIR)**.**

Table 1: Comparisons of the existing integration techniques.

Point-to-Point	**Middleware**
Advantages	
1. The benefit of this integration technique is easy to use when there are small services that needs to be integrated. 2. It is flexible and faster integration approach.	1.Using middleware approach, services do not depend on each other 2.It enables different service to be integrated into the middleware
Disadvantages	
1. This solution does not support scalability, when more service are added, this integration technique becomes unmanageable 2. Another challenge of this technique is that, each service needs to know all the detailed information about the service it wants to integrate with. 3. This technique uses tight coupling and dependence, which result to the major drawbacks as it lacks adaptability and extensibility and does not support dynamic integration and heterogeneous services. 4. This technique lacks scalability and requires immense skilled user-involvement to configure and update the source code whenever there are changes.	1. The challenges with this integration technique is that it is complex to set up. 2. This approach still requires use-intervention whenever there are changes. 3. This technique still uses static integration technique to integrate services, also using this technique services are not able to adapt to the changes. 4. It does not support all the APIs, a developer needs to configure an adaptor that would enable the API to function. 5. This approach does not support dynamic integration as it uses static integration
Spoke and Hub	**ESB**
Advantages	
1. It promotes reusability. 2. It reduces the number of interfaces. 3. It provides more flexibility compare to point-to-point and spoke and hub integration techniques	1. ESB supports the interoperability. 2. This technique is extensible and easy to expand when one needs to connect additional service to their architecture in the future. 3. It supports scalability. 4. It is simple to use, as it supports loosely coupled and reusability of services.
Disadvantages	
1. This technique is not flexible in a high demand of service integration. 2. It is a single point of failure. 3. This technique lacks extensibility, adaptability and does not support dynamic integration. 4. It is limited when it comes to scalability, depending on the box used in the hub. 5. It requires intense user-intervention as it uses static integration.	1. This approach supports only static message routing for service communication. 2. It does not support dynamic integration of heterogeneous services. 3. This is inflexible as it does not support dynamic service integration. 4. The use of static integration may result in scalability problems emanating from multiple integration solutions developed for different purposes.

ISBN: 978-988-14047-5-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

REFERENCES

[1] G W. Shen, Q. Hao, S. Wang, Y. Lia, Y, and H. Ghenniwa, "An agent-based service-oriented integration architecture for collaborative intelligent manufacturing". In: Robotics and Computer-Integrated Manufacturing, pp. 315-325, India, Elsevier Ltd, 2007.

[2] J. Bardin, L. Philippe, and E.Clement, "Towards an Automatic Integration of Heterogeneous services and Devices" Proceedings of 2010 IEEE Asia - Pacific Services Computing Conference. Germany, IEEE, 2013.

[3] M. Makamba, J. Mtsweni, and E.K Ngassam, "Exploring the challenges of dynamic integration of heterogeneous services", Dubai, EIG, 2016.

[4] N. Kaviani, B. Mohabbati, and R. Lea, "ReCoil; A Framework for dynamic Integration of Remote Services-Oriented Components Model", France, ACM, 2011

[5] G. Athanasopoulos, "Interoperability and Heterogeneous Service", China, MCSIN, 2010.

[6] R. Duncan, and D. John, "Dynamic Service Integration for Reliable and Sustainable Capability Provision" Proceedings of 2010 IEEE International conference on system integration and Technologies. London, IEEE, 2010.

[7] Zhu, F. et al., "Dynamic Data integration using web services". UK, IEEE, 2008.

[8] C. Bartelt, T. Fischer, and D Niebuhr, "Dynamic Integration of Heterogeneous Mobile Devices", Germany, ACM, 2012.

[9] C. Reinisch, G. Wolfgang, and F. Praus, "Integration of Heterogeneous Building Automation system using Ontologies". Taiwan, IEEE, 2008.

[10] J. He, N. Chen, and W. Mi, "A Heterogeneous Web service integration method based on Semantic Matching," China, Springer, 2013

[11] RogueWave, Fitting static code analysis into continuous integration, USA: RogueWave Software, 2015.

[12] N. Serrano, J. Hernantes, And G. Gallardo, "Service Oriented Architecture and Legacy systems", UK: IEEE Software, 2015.

[13] J.L Marechaux, "Combining Service-Oriented Architecture and Event Driven Architecture using ESB", USA: IBM, 2006.

[14] Cisco, "Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide", 12.2(25) EW. 1st ed. UK: Cisco, 2009.

[15] S. Asif, and P Webb, "Software System Integration- Middleware -An Overview", Journal of Computer Applications, 121(5), pp. 27-29, 2015.

[16] MuleSoft, "Understanding Enterprise Application Integration - the benefits of using ESB with EAI", U.S.A: MuleSoft, Inc, 2009

[17] M. Makamba, and T. Thinyane, "Integration legacy applications into SOA middleware". Masters in computer science thesis, Alice: UFH, 2012.

[18] A. Goel, "Enterprise Integration EAI vs. SOA vs. ESB", India: IGH, 2011.

[19] Cognizant, Comparing and Contrasting SOA Variants for Enterprise Application Integration, New-Jersey: Cognizant, 2013.

[20] D. Litan, M. Velicanu, and L Copcea (Teohari), 2011. "Business' New Requirement: Infomation system integration methods and technology", International journal of computer and communications, 5(3), pp. 133-145, IEEE, 2011.

ISBN: 978-988-14047-5-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)