2016-11

An overview of existing frameworks for integrating fragmented information systems

Ramadhani, Abswaidi

International Journal of Information Sciences and Techniques (IJIST

10.5121/ijist.2016.6602

AN OVERVIEW OF EXISTING FRAMEWORKS FOR INTEGRATING FRAGMENTED INFORMATION SYSTEMS

Abswaidi Ramadhani[1], Anael Sam [2] and Khamisi Kalegele[3]

[1, 2] School of CoCSE, Nelson Mandela African Institution of Science and Technology,

ABSTRACT

Literatures show that there are several structured integration frameworks which emerged with the aim of facilitating application integration. But weakness and strength of these frameworks are not known. This paper aimed at reviewing these frameworks with the focus on identifying their weakness and strength. To accomplish this, recommended comparison factors were identified and used to compare these frameworks. Findings shows that most of these structure frameworks are custom based on their motives. They focus on integrating applications from different sectors within an organization for the purpose of eliminating communication inefficiencies. There is no framework which guides application's integrators on goals of integrations, outcomes of integration, outputs of integration and skills which will be required for types of applications expected to be integrated. The study recommended further study on integration framework especial on designing unstructured framework which will support and guide application's integrators with consideration on consumer's surrounding environment.

KEYWORDS:

Framework, integration, enterprise, application & system

1. INTRODUCTION

System integration has been in practice for more than 50years ago. It is not a new thing in field of Information Communication and Technology (ICT). It is started when industries tried to integrate functions such as finance, marketing, management and production [1]. To integrate fragmented information systems, framework to guide developers and integrators through out the whole process is needed. According to [2] framework is a "basic structure underlying a system, concept, or text". In his paper on desirability of integration, a scholar [3] detailed that "integration of separate and isolated islands of systems began with projects which involved few applications". Enterprises and organizations are the one which instigated integration after rising of several integration frameworks which included programming interface, Middleware and Enterprise Resource Planning (ERP) just to mention few. In earlier period, systems were designed and developed without worries in change of technology. It was not developed without consideration of been integrated in future. Every developer estimated the software requirements based on existing situations. As time goes with advance in technologies such as internet and concept of e-commerce, enterprises and big companies started to integrate customers and business processes with their internal systems [4]. There are different reasons and requirements which contributed to

DOI : 10.5121/ijist.2016.6602

the needs of application integration [5]. Among of these requirements were to increase ability of having information which flows inside enterprises and organizations from different applications which were built to respond to different requests. Another requirement were the fact that existing applications are providing functionalities in an isolated mode [6, 7]. Also it was hard to share data and functionality offered by different application in an efficient and unified ways [8]. In other hand, literatures shows that some business had merged but the reality is that they merge with the aim of cutting redundancies and lowering the cost of services they offer. The same reason applied to application integration since many companies have diverse applications which run in different platforms for different purpose.

As time goes, it reached the point where companies and sectors appreciated the important of integration. It realized that in order to be able to save their stakeholders such as customers, suppliers, employee and partners requirements, they have to unify their applications to provide a common access point for all services they offer. Some scholars including [9] advocate that, unification of applications are needed so as to be able to provide services that originated from diverse sources. This unification involves integrations of data and business process between diverse applications. Not only those days but even today's business applications are rarely live in isolation. Literatures show that there are several frameworks which emerged with the aim of facilitating application integration. Some of these frameworks are programmed interface, Enterprise Resource Planning (ERP), Middleware, Enterprise Application Integration (EAI), Web service, Integration Platform as a Service (iPaaS), Electronic Data Interchange (EDI), shared database and Enterprise Service Bus (ESB). These frameworks were developed with intention of facilitating systems and applications integration. Despite the fact that there are many developed framework but their analysis in terms of weakness and strength are not well known. So this paper was aimed at reviewing these frameworks with the focus of identifying its weakness and strength.

1.1 SCOPE OF THE STUDY

The study concentrated on nine integration frameworks which were programmed interface, Enterprise Resource Planning (ERP), Middleware, Enterprise Application Integration (EAI), Web service, Integration Platform as a Service (iPaaS), Electronic Data Interchange (EDI), shared database and Enterprise Service Bus (ESB). There were not real system or data integration taken place instead the comparison frameworks were identified and used to compare all identified integration framework technologies.

1.2 METHOD AND METHODOLOGY

According to [10] research method is a real research tool used or component of research. For example, a qualitative method such as interviews is called research method. Methodology is the justification for using a particular research method. Here methodology was defined as a reason for using a certain method in research. Other scholar [11] elaborated that research methods are the various procedures, schemes and algorithms used in research. All the methods used by a researcher during a research study are termed as research methods. Research method includes theoretical procedures, experimental studies, numerical schemes and statistical approaches. This scholar added that research methodology is a systematic way to solve a problem. It is a science of studying how research is to be carried out. Essentially, the procedures by which researchers go about their work of describing, explaining and predicting phenomena are called research methodology. Research methodology was also defined as the study of methods by which

knowledge is gained. To accomplish the task of reviewing existing framework for integrating fragmented information systems, the comparison factors were identified through literature review and used to compare these frameworks. Findings were obtained and presented in last chapter of this study.

2. LITERATURE REVIEW

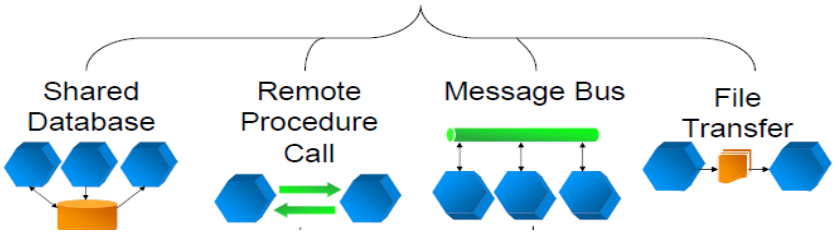
2.1 FRONT-ENDED AND BACK-ENDED INTEGRATION

In integration, applications or systems can either be front-end or back-end integrated. Front-end integration is the type of integration in which two or more applications are integrated at client side [12]. Front-ended integration solution focuses on creating a unified user interface. It takes place at the presentation layer which is an interface between user and data access layer. Back-ended integration is type of integration in which applications are integrated at server side. Its integration takes place at data access layer. Codes for this integration are written in server. Back-end integration involves uses of some orchestration to accomplish it.

2.2 STYLES OF APPLICATION INTEGRATION

Application integration involves several styles because integration needs is not the same. If integration needs were the same, there would be only one integration style. In reality, application integration involves a sort of considerations and consequences that should be taken into account for any integration work [13]. In some extents, integration becomes a vital to many fields due to fact that even simple project has numerous applications that needs to work together to provide a unified service. We can avoid application integration if and only if we can develop a single application that doesn't need collaboration with any other applications. Some of decision criteria which should be considered and which are the main source of integration styles are; application coupling, intrusiveness, technology selection, data timelines, remote communication, data format, reliability, data format evaluation and extensibility. When developer and integrator thinks about integrating applications, they should also think about minimizing integration codes as well as changes to the applications as emphasized in intrusiveness of integrated applications. Coupling emphasizing minimization of dependencies among integrated applications. Technology selection results into different integration style because different integration techniques require different software and hardware [14]. These tools can be expensive and lead to vendor lock-in and it can increase the learning curve for developers. Another consideration which produces integration style is data format. Integrated applications must agree on format of data they exchange. It can be either difficult or impossible to change existing applications to use a unified data format. Data format evolution and extensibility looks at how the agreed format can change over time and how that change will affect the applications. Integration style also attributed by data timelines, this looks at minimization of time taken when one application decides to share some data and other applications have that data. Data or functionality looks at possibility of sharing data or functionality. Integration considers remote communication in integration. Here involves synchronous and asynchronous. Synchronous is when procedure waits while its sub-procedure executes while asynchronously is when we start sub-procedure while procedure continuing with its own processing simultaneously. Application integration also considers reliability of the targeted service since remote connections are slow and less reliable than local function call. The above explanation shows that there are many diverse criteria which must be considered when we

choose and design integration methods. So in answering the question "Which integration approach best addresses which of these criteria?" here is when we come up with different integration style because there is no single integration method which addresses all criteria. To overcome this, multiple integration styles have been involved over time. Figure 1 shows those application integration styles [15].



Application Integration styles: Adopted from Hohpe (2003)

Figure 1: Application Integration Styles

According to [16], In file transfer integration style, each application generates files of shared database for other applications to consume and consume files that other applications have generated. Integrators take the responsibility of transforming files into different formats. The formats should be understandable to other applications. These files should be produced at regular intervals according to the nature of the business. In a shared database application integration style, all application stores data they wish to share in a common database. In this style, integrator integrates applications by making them to store their data in a single shared database which can be accessed by all participated applications and define the schema of the database to handle all the needs of the different applications. Shared database is easier to adopt when we make use of SQL-based relation databases which is ease and almost all application platform works with it. Integration of applications which use SQL has no need of multiple file formats [17].

According to scholar [18], remote procedure call integration involves creating mechanism for one application to invoke a function which is resides in another application and passing the data that needs to be shared and invoking the function that tells the receiver-application how to process the data. Accomplishing this integration could involve developing applications as a large component with encapsulated data and provide an interface to allow other applications to interact with the running application. So each application exposes its procedures so that they can be invoked remotely. Messaging is among of the application integration style in which each application connect to a common messaging bus system to exchange data. The applications uses message to transfer packets of data reliably, frequently, immediately and asynchronously, using customizable formats. Messaging application integration services enables applications of all sizes and capabilities to connect and exchange messages in a standard format. This style accomplished through routing, data mapping and file format transformation with an aim to facilitate business transactions.

3. OVERVIEW OF EXISTING INTEGRATION TECHNOLOGIES

History of integration framework was carried in concept of enterprise integration as integration was initial started in enterprises. The idea of Enterprise Integration (EI) has been there since the

early day of computers in industry but manufacturing industry used the term Operation Integration (OI) to mean EI. EI was official documented in 1950s, at these time systems were integrated by using programmed interface. These integrations were point to point in nature. Difficulties of programmed interface and change in technologies stepped EI into another step called shared databases in the year of 1960s. This technique provided direct access to common data from different locations. Another technique called Electronic Data Interchange (EDI) was emerged in 1970s [19]. This technique enabled separated business enterprises to integrate with each other using standard interfaces which are predefined. EDI was followed by Enterprise Resource Planning (ERP) in 1980s. ERP used single database to integrate all applications which intended to be integrated. Another generation of EI was called Middleware. This generation was emerged in the 1990s and used a special layer for integrating different applications and databases. In that period, a language called XML was discovered and used as a standard language for web communication. This language was used in middleware integration. The concept of Enterprise Application Integration (EAI) in the field of EA was introduced in 1995. This technique employee software and computer systems' architectural principles to integrate set of enterprise applications. Through EAI, developers managed to integrate legacy systems and different web applications [7].

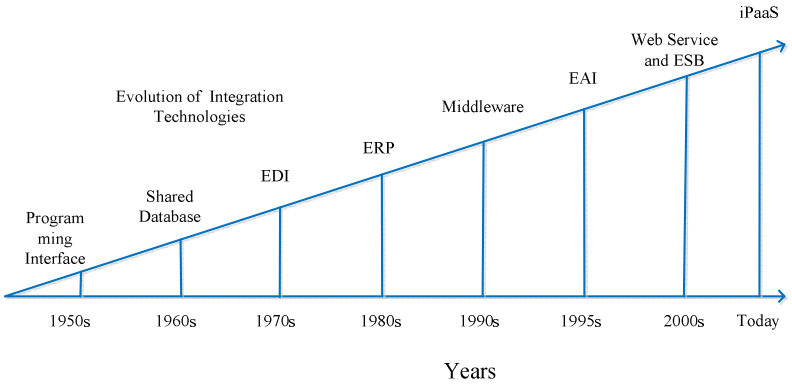


Figure 2: Evolution of Integration Technologies

Web Service was introduced in 2000s and became a primary method for systems integration. It communicates over WWW using HTTP. Web service provides a standard interoperating among applications which runs in different platforms and frameworks. Through web service, developer managed to integrate not only internal applications but also external service provider. ESB which introduced in 2002 provided ability to integrate service in centralized entity. Everything was connected to ESB and ESB routed data. Despite the facts that ESB increased scalability and flexibility, it also slowed communication time and increased overhead [20]. In system integration, iPaaS is the latest integration method. It allows enterprise to integrate the whole IT infrastructure via software as service architecture. IPaaS delegates all complexity to third vendor rather than hosting an integration framework in house. This architecture reduces maintenance overhead, provides instant access to latest product features and ensures extended resources availability. IpaaS operates as a complete integration platform. In contrast from the past enterprise integration, iPaaS included adapters for different applications, data platform and operating systems that allow creating of integration points without developing any custom code [21].

3.1 OVERVIEW ON PROS AND CONS OF EXISTING FRAMEWORK TECHNOLOGIES

According to [22], pros and cons are advantages and disadvantages which user of a certain framework should careful consider before making sensible decision. Existing frameworks also has pros and cons which users can use as among of factors during making decision on what framework to use.

Framework	Pros	Cons
Shared DB	✓ It offer centralized resource, so easy to maintenance and upgrade. ✓ It serves as common platform for all partners if it implemented right. ✓ Platform and data can be reused. ✓ It provides common connectivity and resource pooling	✓ Any changes to the DB or data might require all the partners to sign off ✓ Standardization is required as it share across many applications ✓ Governance is required to make sure only the intended data is stored in the DB. ✓ Expansion of features can be complex once it implemented in a particular standard
EDI	✓ High speed data transmission ✓ Automated Data entry ✓ No human intervention is involved so minimal error ✓ Automatically data Validation ✓ Lower administrative, resource and maintenance cost. ✓ Faster processing compare to traditional	✓ Too expensive in setup and maintenance of some formats of EDI ✓ Time consuming in initial setup ✓ It needs electronic protection against virus ✓ Limits trading partners to only those who use EDI ✓ It needs proper backup
Web Service	✓ Interoperability: It works outside of private networks ✓ Usability since it allow business logic of many systems to be exposed over the Web	✓ Its core protocols like HTTP and HTTPS are simple but not mean long term session. ✓ Its protocols HTTP and HTTPS are stateless
ESB	✓ It is lightweight since it is made up of many interoperating services ✓ Easy to expand ✓ It is scalable and distributable since its functionality can be dispersed across a geographically distributed network	✓ Single point of failure ✓ It is difficult to get good developer with ESB skills and substantial experience ✓ Risk of regression. Since all applications are talking via a centralized entity, any change to the entity bears a risk of introducing a regression.
ERP	✓ It is visible to all important management personnel ✓ Automatic and logical workflow from one department/function to another ✓ It offer unified and single reporting system to analyze the statistics/status ✓ It is more secure as centralized security policies can be applied to them ✓ It make easy for order, inventory and revenue tracking	✓ High cost of software planning, customization, configuration, implementation and testing ✓ It take long time up to 1-3years from deployment to get complete fully functional ✓ Migration of existing data to the new ERP systems is difficult (or impossible) to achieve. ✓ It is difficult to archive in decentralized organizations
Middleware	✓ It allow real time information access among systems ✓ Maintains information integrity across multiple systems ✓ It helps developer to create networked applications	✓ Higher cost of development ✓ There are few people with experience on middleware in the market ✓ Too many platforms to be covered ✓ It threatens the real-time performance of a system
EAI	✓ It speedup transactions at reduced cost ✓ It enable information sharing ✓ It enable process automation	✓ It form single point of failure as it use central engine ✓ Their implementation is expensive

Table 1: Pros and cons of existing frameworks

4. COMPARISON FACTORS OF INTEGRATION TECHNOLOGIES

With the focus of this study i.e. to review the existing frameworks for integrating disparate applications, it is crucial to compare these integration technologies with integration requirements factors. There are many scholars who worked on comparison factors. One of these scholars is [23]. Thermistocleous suggested several factors to consider during comparing integration technologies which includes maintainability, flexibility, scalability, portability, reusability and maturity. Another scholar [24] pointed out that, when we compare integration technologies, we have to consider factors like implementation, pricing, portability, tools and servers, promoting companies and maturity of platform. Other scholar called [25], advocated that during comparing and contrasting integration framework technologies, we should consider factors like open source, basic concepts or architecture, testability, deployment, popularity, commercial support, IDE-Support, error handling, monitoring, enterprise readiness, Domain Specific Language (DSL), number of components for interfaces, usability, maintainability, community, enterprise support, functionality, flexibility, connectors, cost, licensing and expandability. According to [26], the best comparisons factors in integration technologies are scalability, lightweight, coupling and security mechanism.

For this study, we used maintainability, flexibility, coupling, lightweight, scalability, security mechanism, portability, reusability and maturity as it supported by [27]. Bad and good connections of internet were added here due to context of the study area which is Tanzania. Tanzania is characterized by good and bad internet connections (TCRA, 2015). All these comparison factors are based on characteristics of good integration frameworks which represents supports which will be provided when an organization adopt it. Detailed description of each factors are presented under this section plus comparisons matrix in table 3.

Factors	Descriptions
Maintainability	[27] recommended that maintainability is an essential factor in integration technologies. It is the probability of performing a successful repair action within a given time. It measures the ease and speed with which a system can be restored to operational status after a failure occurs [28]. Integration technologies should lead to product which is easy to maintain.
Flexibility	[29] advocated that flexibility in modification and flexibility in functionality explains the capability to operate in different environment as well as capability to respond to rapid adjustment with minimal efforts. In the context of integration technologies, flexibility supports both flexibility in modification and functionality.
Scalability	[30] suggested that scalability is an important character in system software. It refers to ability of the information system to handle growing amount of work when placed upon it. In integration technologies, the adopted technology should be scalable enough.
Coupling	According to [31] , coupling is the degree to which software components are dependant to each other. In a loosely-coupled architecture, component remains independent and allows middleware software to manage all communications between them while in tight coupled architecture, codes must be executed to allow communication[32]. In context of integrating systems, we expect loose coupled technology.
Lightweight	Lightweight means anything that is relatively simpler or faster or that has fewer parts than something else [33]. In integration technologies, we expect to adopt lightweight framework technology.

Portability	[33, 32] supported that, portability allows working in multiplatform and multi environment. In integration technology, we expect technology which will enable to integrate system developed by using different platforms.
Security mechanism	[34] recommended that, security in context of information technology is the defence of systems or applications against intrusion and unauthorized use of resources. In integration frameworks, we expect framework which is self defensive or one which has inbuilt mechanism for self defensive.
Reusability	According to [35], reusability is the capability of using existing components or software solutions to build new applications. It is essential in application integration as it reduces the implementation time and cost. It results in more flexible, manageable and maintainable integrated systems.
Maturity	According to [27], maturity shows whether an integration technology is mature or not. The more mature an integration technology, the better it is. The reason for this is that analysts and developers trust more mature technologies than immature one
Good & Poor Internet Connection	In case of good and poor internet connection, we concentrate on applicability of the integration technologies in area with good internet connection and in area with poor internet connection. We expect integration framework which is applicable in all areas.

Table 2: Description of comparison factors

4.1 Comparison Matrix

FRAMEWORK TECHNOLOGIES	CRITERIA												
	Loose Coupling	Maturity	Security Mechanism	Tight Coupling	Portability	Reusability	Scalability	Good Internet	Poor Internet	Maintainability	Lightweight	Flexibility	Reference
Programmed Interface	x	x	X	√	x	x	x	√	√	x	x	x	(Cognizant, 2013, De Leusse, 2007)
Shared Database	x	√	X	√	x	x	x	√	√	x	x	x	(Cognizant, 2013, De Leusse,2007), (Stater, 2010), (Bhaskar, 2001), (Khan, 2015)
EAI	x	√	X	x	x	x	x	√	x	x	x	x	(Bhuvaneswari,2011),(Khan,2015),(Chakraram,2016),(Cognizant, 2013)
ERP	x	√	X	√	x	x	x	√	x	x	√	x	(Kumar, 2014), (Zhang, 2003), (Cognizant, 2013) (Palmquist,2014), (Sadera,2015),
EDI	x	√	X	√	x	x	x	√	x	x	x	x	(Guzik, 2014), (Ray,2010), (Sparrow,2016),(Magutu, 2010),(John, 2002)
Middleware	x	x	X	x	x	√	x	√	x	x	√	x	(Gordon, 2004), (Stater, 2010), (Chapin, 2002) (Bhaskar,2001),(Kumar, 2014), ,(Schmidt,2003)
Web Services	√	√	√	x	√	√	√	√	x	√	√	√	(Cognizant, 2013), (Chakraram,2016), (Merritt, 2012),(Claudera, 2016),(Feraga, 2011), (Srinath,2014),(Kamilaris, 2009), (Saab,2007), (Hmida,2005),(Cellary,2010),(Panhelainen,2008) , (Gharzouli,2002)
ESB	√	√	√	x	√	√	x	√	x	√	√	x	(Murakani,2013),(Edmorris,2010),(Feraga, 2011), (Thayaparan,2011),(Merritt, 2012), (Singh,2012)(Vanderveck,2014),(Ahson,2011), (Saab,2007),(Cognizant, 2013),
iPaaS	√	√	X	x	√	√	√	√	x	√	x	√	(Murakani,2013), (Ovum, 2015), (Gartner, 2011), (Merritt, 2012),(Annenko,2016), (Saab,2007)

Table 3: Comparison Matrix

The criteria used in this comparison are based on structured framework. It shows the properties which will be beard by systems which will be integrated by a certain framework. Generally, criteria and framework based on eliminating the problems of interacting between application rather than guiding the integrator during applications integration. Among all used and identified frameworks, there is no one which is a set of fact or ideas for guiding integration. All these structure framework based on eliminating integration problem.

5. GENERAL OBSERVATION AND RECOMMENDATION FOR FURTHER STUDY

5.1 GENERAL OBSERVATIONS

Based on review made on existing frameworks, it had been revealed that most structure framework is custom based on their motives. For instance EAI focuses on integrating applications from different sectors within an organization for the purpose of eliminating communication inefficiencies, automating straightforward processes, or addressing vendor independence challenges while ESB focuses on integrating applications over a bus like infrastructures [36]. There is no framework which guides application's integrators on goals of integrations, expected outcomes, expected outputs and skills which will be needed for the job. Also existing frameworks does not put into consideration the environment of the expected users or customers of the integrated systems. Some customers are living in area with insufficient internet connections while other are living in area with good internet connections, comprehensive framework should put all this into considerations. Absence of these considerations into existing framework had contributed to increase the problem of inaccessibility of farmer's information in Tanzania despite the facts that there are many projects which had been initiated to solve this problem. For example information contained by Kariakoo Market Corporation (KMC), Livestock Identification and Traceability System (LINKS) and First Mile Project can't be accessed by customers via USSD while information in Tigo Kilimo can't be accessed through web interface. In Agriculture Routine Data System (ARDS), farmers have to go to the responsible officers to request this statistics information. There is a need of finding alternative ways of designing framework which will accommodate this and solve the existing gap of lack of framework which is set of idea or facts which guides integrator in integrating applications depends on scenario of applications itself.

5.2 GENERAL RECOMMENDATION FOR FURTHER STUDY

Based on the observed problems in the existing frameworks, this study calls for further study on designing of new framework. Framework which will explore factors surrounding the expected consumers. These factors can either be those which influence or hinder integration of applications at various layers with the landscape of Agro-information in Tanzania context and other related countries. The new framework should organize and put into consideration all these factors and provide guidelines for integration of applications at various layers. Moreover, new framework should be able to guide integrators on goals of integration, expected outcomes of integration, expected output of integration and skills which will be needed for the whole job of that applications selected for integration.

REFERENCES

[1] Alsene, E., Computerized integration and the organization of work in enterprises. Int'l Lab. Rev., 1994. 133: p. 657.

[2] Sharma, V., Framework for multi-protocol label switching (MPLS)-based recovery. Framework, 2003.

[3] Singletary, L.A. Applications integration: is it always desirable? in System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on. 2004: IEEE.

[4] Erasala, N.Y., David C Rajkumar, TM, Enterprise Application Integration in the electronic commerce world. Computer Standards & Interfaces, 2003. 25(2): p. 69-82.

[5] Bagherinia, S., Open Source Enterprise Service Bus: Analyzing impact and value in a large organization. 2013.

[6] Yang, B.L., Lu, A Case Study of Enterprise Application Integration Based on Workflow Management System, in Research and Practical Issues of Enterprise Information Systems II. 2007, Springer. p. 425-431.

[7] Linthicum, D.S., Enterprise application integration. 2000: Addison-Wesley Professional.

[8] Dutka, Ł.S., Renata Wrzeszcz, Michał Król, Dariusz Kitowski, Jacek, Uniform and efficient access to data in organizationally distributed environments, in eScience on Distributed Computing Infrastructure. 2014, Springer. p. 178-194.

[9] Fazlollahi, A., Benefits of Enterprise Integration Systems. 2012.

[10] Rajasekar, et al., Research methodology. arXiv preprint physics/0601009, 2006.

[11] Rexford, N. and P. Cunnington, Methodology/Dress History/Historiography.

[12] Rouse, M. front-end and back-end integration. 2016 [cited 2016 August 26]; Available from: http://searchdatacenter.techtarget.com/definition/EDI.

[13] Briggs, D.J., A framework for integrated environmental health impact assessment of systemic risks. Environmental Health, 2008. 7(1): p. 1.

[14] Klemm, et al., Framework for integrating existing and new information technology applications and systems. 2006, Google Patents.

[15] Gould, et al., Techniques for performing a remote procedure call using remote procedure call configuration information. 2013, Google Patents.

[16] Pautasso, et al. Restful web services vs. big'web services: making the right architectural decision. in Proceedings of the 17th international conference on World Wide Web. 2008: ACM.

[17] Bos, et al., From shared databases to communities of practice: A taxonomy of collaboratories. Journal of Computer-Mediated Communication, 2007. 12(2): p. 652-672.

[18] Gould, et al., Techniques for performing a remote procedure call using remote procedure call configuration information. 2014, Google Patents.

[19] Leiner, B.M., Vinton G Clark, David D Kahn, Robert E Kleinrock, Leonard Lynch, Daniel C Postel, Jon Roberts, Larry G Wolff, Stephen, A brief history of the Internet. ACM SIGCOMM Computer Communication Review, 2009. 39(5): p. 22-31.

[20] Wiederstein, et al., ProSA-web: interactive web service for the recognition of errors in three-dimensional structures of proteins. Nucleic acids research, 2007. 35(suppl 2): p. W407-W410.

[21] Jacome, A. integration Platform as a Service: The Latest Link in Evolution of Information Systems Integration. 2014 [cited 2016 August 6]; Available from: https://www.linkedin.com/pulse/20140610191903-30466066-ipaas-the-latest-link-in-evolution-of-information-systems-integration.

[22] Forman, et al., Decision by objectives: how to convince others that you are right. 2001: World Scientific.

[23] Thermistocleous, M.G., Evaluating the adoption of enterprise application integration in multinational organisations. 2002, Brunel University, School of Information Systems, Computing and Mathematics.

[24] Gunjan Samtani, D.S., Web Services and Application Frameworks, (.NET and J2EE) http://www.nws.noaa.gov/oh/hrl/hseb/docs/ApplicationFrameworks.pdf. 2010.

[25] Wähner, K. Which Integration Framework Should You Use – Spring Integration, Mule ESB or Apache Camel? 2016 [cited 2016 September 15]; Available from: https://dzone.com/articles/which-integration-framework.
[26] Cognizant, Comparing and Contrasting SOA Variants for Enterprise Application Integration, https://www.cognizant.com/InsightsWhitepapers/Comparing-and-Contrasting-SOA-Variants-for-Enterprise-Application-Integration.pdf. 2013.
[27] Khoumbati, K., Themistocleous, Marinos, Irani, Zahir. Integration technology adoption in healthcare organisations: A case for Enterprise Application Integration. in Proceedings of the 38th Annual Hawaii International Conference on System Sciences. 2005: Ieee.
[28] Stromgren, C., Terry, Michelle, Cirillo, William, Goodliff, Kandyce, Maxwell, Andrew. Design and Application of the Exploration Maintainability Analysis Tool. in AIAA Space 2012 Conference and Exposition, AIAA-2012-5323, Pasadena, CA. 2012.
[29] Kumar, A process model for analyzing and managing flexibility in information systems. European Journal of Information Systems, 2014. 23(2).
[30] Putnik, G.S., A. ElMaraghy, H. Teti, R. Koren, Y. Tolio, T. Hon, Scalability in manufacturing systems design and operation: State-of-the-art and future developments roadmap. CIRP Annals-Manufacturing Technology, 2013. 62(2): p. 751-774.
[31] Wigmore, I. monolithic architecture. 2016 [cited 2016 August 28]; Available from: http://whatis.techtarget.com/definition/monolithic-architecture.
[32] Rouse, M. Semantic integration: Loosely coupling the meaning of data. 2011 [cited 2016 August 29]; Available from: http://searchnetworking.techtarget.com/info/problemsolve/Cloud-Networking.
[33] Lee, J.-K., Lightweight alarm manager on web browser and service method thereof, and method of providing alarm information therefor US 20040210420 A1. 2004.
[34] Hau, D., Global Information Assurance Certification Paper, https://www.giac.org/paper/gsec/3161/unauthorized-access-threats-risk-control/105264. 2013.
[35] Singh, S., Singh, Rishideep, Reusability Framework for Cloud Computing. arXiv preprint arXiv:1210.8011, 2012.
[36] Tello, M., Scalable integration and pre-processing of sensor data stream. 2014.